UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MATEUS DA SILVA SOUSA DIAS

USANDO LINKED OPEN DATA PARA GERAÇÃO DE EXPLICAÇÕES EM SISTEMAS DE RECOMENDAÇÃO MUSICAL

PATOS – PB
2020

MATEUS DA SILVA SOUSA DIAS

USANDO LINKED OPEN DATA PARA GERAÇÃO DE EXPLICAÇÕES EM SISTEMAS DE RECOMENDAÇÃO MUSICAL

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração: Inteligência artificial

Orientador: Professor Dr. Ricardo Santos de Oliveira

PATOS – PB

2020

D541u Dias, Mateus da Silva Sousa.

Usando linked open data para geração de explicações em sistemas de recomendação musical [manuscrito] / Mateus da Silva Sousa Dias. - 2020.

60 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas , 2020.

"Orientação : Prof. Dr. Ricardo Santos de Oliveira , Coordenação do Curso de Computação - CCEA."

1. Sistemas de recomendação. 2. Filtragem colaborativa. 3. Linked Open Data. 4. Dbpedia. I. Título

21. ed. CDD 005.3

Elaborada por José E. da S. Eugênio - CRB - 15/591 BSC7/UEPB

Mateus da Silva Sousa Dias

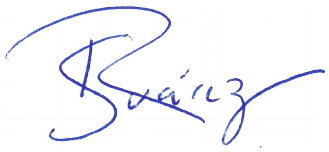
USANDO LINKED OPEN DATA PARA GERAÇÃO DE EXPLICAÇÕES EM SISTEMAS DE RECOMENDAÇÃO MUSICAL

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Bacharel em Ciências da Computação.

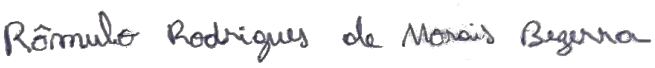
Aprovado em 26/11/2020

BANCA EXAMINADORA

Prof. Ricardo Santos de Oliveira
(Orientador)



Prof. Pablo Ribeiro Suárez
(Examinador)



Prof. Rômulo Rodrigues de Morais Bezerra
(Examinador)

AGRADECIMENTOS

A Deus pelo dom da vida, pela concessão do conhecimento e pelo suporte para conclusão deste trabalho.

Aminha família, que sempre me apoio em momentos difíceis. Sem eles, seria impossível chegar até aqui.

Aos meus pais que nunca deixaram faltar nada para que eu concluísse meus estudos, sempre me incentivaram e me apoiaram.

A minha esposa, Maria, que sempre foi paciente e esteve ao meu lado nas dificuldades.

A minha irmã, Cleonides, que sempre foi um exemplo de superação diante das dificuldades que a vida impõe.

Ao meu cunhado, Renê que, além de ser um exemplo, me apoiou e auxiliou neste trabalho.

Ao meu orientador, Ricardo, pelo desafio oferecido e disponibilidade, além de sua paciência e conhecimento fornecido. Sem ele, este trabalho não estaria concluído.

Aos amigos que consegui na faculdade. Cada um deles me auxiliou para que eu chegasse até aqui.

A todos os meus professores e funcionários da UEPB. Cada um de vocês tem um papel importante para que todos os alunos alcancem seus objetivos e todos são merecedores de gratidão.

“(...) Aprende que as circunstâncias e os ambientes têm influência sobre nós, mas nós somos responsáveis por nós mesmos.
(...) Portanto, plante seu jardim e decore sua alma, ao invés de esperar que alguém lhe traga flores. (...)”

Veronica Shoffstall

RESUMO

A utilização de sistemas de recomendação está crescendo de forma exponencial. No passado, sua utilização se restringia apenas a algumas áreas, como a de entretenimento, por exemplo. Contudo, atualmente os sistemas de recomendação têm ganhado novos horizontes e sua presença é notada cada vez mais, especialmente em áreas mais cruciais. Ao passo que a qualidade da recomendação está sendo aprimorada constantemente e levando em consideração a presença de sistemas de recomendação em áreas que afetam diretamente a vida de usuários ou terceiros, a interpretabilidade se mostra cada vez mais essencial em sistemas de recomendação. O objetivo desta pesquisa é o desenvolvimento de um sistema de recomendação musical, baseado na abordagem de filtragem colaborativa. Além disso, o sistema também tem como objetivo a geração de justificativas para as recomendações. Esse tema aborda a questão da interpretabilidade, que se mostra cada vez mais importante em sistemas de recomendação. Apesar de não conseguir gerar justificativas para todas as recomendações, o sistema desenvolvido conseguiu alcançar uma taxa de 94,72% de justificativas geradas. O alcance em questão se mostrou satisfatório para o objetivo, levando em consideração a ausência de alguns dados nas bases extraídas do site DBpedia.

Palavras-chave: WWW, RDF, Linked Data, Linked Open Data, Dbpedia, Sistemas de Recomendação, Filtragem Colaborativa, Interpretabilidade.

ABSTRACT

The use of recommendation systems is growing exponentially. In the past, its use was restricted to only a few areas, such as entertainment, for example. However, nowadays recommendation systems have gained new horizons and their presence is increasingly noticed, especially in more crucial areas. While the quality of the recommendation is constantly being improved and taking into account the presence of recommendation systems in areas that directly affect the lives of users or third parties, interpretability is increasingly essential in recommendation systems. The objective of this research is the development of a musical recommendation system, based on the collaborative filtering approach. In addition, the system also aims to generate justifications for the recommendations. This theme addresses the question of interpretability, which is increasingly important in recommendation systems. Despite failing to generate justifications for all recommendations, the developed system managed to achieve a rate of 94.72% of justifications generated. The scope in question proved to be satisfactory for the objective, taking into account the absence of some data in the databases extracted from the DBpedia website.

Keywords: WWW, RDF, Linked Data, Linked Open Data, Dbpedia, Recommendation Systems, Collaborative Filtering, Interpretability.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

Segundo Whitehead (2007 apud VIEIRA, 2014), nos últimos anos o advento da World Wide Web (WWW), criada por Tim Berners-Lee, reinventou a forma de comunicação e compartilhamento de informações. Feitos que antes seriam mais difíceis, como a comunicação instantânea entre duas ou mais pessoas ao redor do mundo, nos dias atuais tornou-se algo comum. Além da comunicação interpessoal, a WWW trouxe também novas possibilidades de compartilhamento de informações em forma de documento, considerando que o principal objetivo da Web seria a conexão de documentos online.

Porém, apesar da quantidade de benefícios que a criação da World Wide Web trouxe ao mundo, as diretrizes que regiam a Web, tornando-a uma grande central de compartilhamento de informações, não foram aplicadas aos dados estruturados (BIZER; HEATH; BERNERS-LEE, 2011). Durante muitos anos a vinculação de coisas disponíveis online só foi aplicada aos hipertextos, multimídias e hipermídias, causando um desperdício de aproveitamento que os dados vinculados à Web poderiam trazer.

Contudo, nos últimos anos, o quadro que retratava apenas a vinculação de informações na Web foi remodelado. Despertou-se o interesse em explorar a importância da conexão de não apenas informações, mas também dados estruturados que poderiam ser lidos por máquina (BIZER et al., 2008). A isso deu-se o nome de “Linked Data” ou “Dados Vinculados”. O termo faz referência a um grupo de práticas que servem como recomendações para publicações de dados vinculados na Web. O uso dos princípios de Linked Data nos leva a uma quantidade consideravelmente alta de novas aplicações que poderiam se aproveitar do projeto.

Relacionado ao projeto Linked Data existe o projeto Linked Open Data (LOD), considerado por Jain et al. (2010) uma pedra angular na implantação do que é proposto no Linked Data. A principal diferença entre as duas propostas é que o Linked Data, como o próprio nome diz, é a conexão de dados, mas não necessariamente dados abertos ao público. Pode-se utilizar como exemplo dados pessoais de grandes organizações privadas sobre vendas ou dados sobre pesquisa de mercado, onde nesse caso apenas a organização interessada tem acesso a tais informações (W3, 2006). Já o objetivo do projeto LOD é rastrear dados livres que se encontram disponíveis para acesso público, dispersos na Web e republicá-los online

mas, dessa vez, links seriam implantados para realizar a associação entre esses dados. Tais dados não devem ter restrições de acesso e, dessa forma, qualquer que seja a entidade (usuário comum, organizações públicas ou privadas, público em geral) que tente acessar esses dados não teria nenhum tipo de impedimento, tornando as informações abertas para uso público.

1.1. Problemática

Para Bizer et al. (2009), as bases de conhecimento estão sendo essenciais para o aperfeiçoamento da inteligência de pesquisa na Web. Mais que isso, essas bases estão sendo fundamentais para a implantação dos princípios de LOD, ajudando cada vez mais a vincular dados na rede. O maior exemplo que pode ser citado de base de conhecimento é o projeto DBpedia. Fundado a partir de um esforço da comunidade crowdsourcing, ou seja, pela contribuição de um grande número de pessoas que atuam em prol de um mesmo objetivo, o projeto extrai informações da Wikipédia e as disponibiliza em forma de dados estruturados. É visto que a Wikipédia disponibiliza apenas uma forma de pesquisa em texto corrido, o que dificulta a extração de informações mais enriquecidas ou específicas. Realizar pesquisas para obter informações mais detalhadas, como por exemplo, saber a população total de alguns países do sul da Europa seria uma tarefa difícil tendo em vista as formas de pesquisa disponíveis na Wikipédia. Entretanto, com a base de conhecimento construída pelo projeto DBpedia, algumas poucas linhas de comando iriam retornar a informação desejada (LEHMANN et al., 2015; AUER, 2007 et al.).

A Wikipédia é o maior exemplo de enciclopédia criada de forma colaborativa no mundo (BIZER et al., 2009). Utilizando-se dessa base de conhecimento, é possível desenvolver inúmeros aplicativos com finalidades variadas, tendo em vista que as informações da Wikipédia são atualizadas frequentemente, já que a base de conhecimento é mantida por milhões de colaboradores ao redor do mundo (BIZER et al., 2009).

Os sistemas de recomendação, como o próprio nome já diz, servem para recomendar assuntos, produtos, vídeos e músicas, de acordo com o interesse do usuário. A recomendação é feita com base em itens que o usuário se interessou anteriormente, ou em comparação com outros usuários de características semelhantes. Em alguns casos, os métodos de filtragem do sistema de

recomendação são mesclados e assim, cria-se um sistema de recomendação híbrido (VIEIRA e NUNES, 2012).

Porém, um problema que atinge quase todo sistema de recomendação é a razão da oferta de produtos e serviços dos usuários. No dia-a-dia, durante o uso da internet, seja nas redes sociais, realizando uma pesquisa simples em um buscador, utilizando um streaming de vídeos, filmes ou músicas nos deparamos frequentemente com recomendações sobre produtos (no caso de utilização das redes sociais ou no buscador), vídeos sobre algum assunto específico, filmes sobre um determinado gênero ou músicas de estilos diferentes. Dentre todo esse universo de opções que nos são oferecidas, alguns poucos sites fornecem o motivo da recomendação. Todo o mais é ignorado pelo usuário, onde muitas vezes ele não sabe o porque daquela recomendação específica ter sido feita a ele.

Sistemas de recomendação musical podem utilizar essas informações úteis extraídas da Wikipédia para montagem de grafos, com o objetivo de formar algoritmos cada vez mais precisos, além de, por meio disso, gerar justificativas para o usuário da recomendação dada. Por meio dos grafos é possível relacionar os gêneros musicais e, com isso, gerar uma justificativa para o usuário. Por exemplo, com a aplicação dos grafos em sistemas de recomendação consegue-se relacionar gêneros e subgêneros, quais instrumentos musicais são utilizados em determinados gêneros ou se a origem estilística de um gênero é a mesma que de outro. Relacionando essas informações e aplicando-as aos gêneros musicais, é possível gerar justificativas para as recomendações.

Segundo Molnar (2018), a interpretabilidade não tem definição matemática. Entretanto, Miller (2018) define interpretabilidade como o grau que um ser humano é capaz de entender uma decisão. Molnar (2018) complementa explicando que quanto maior a interpretabilidade de um modelo de aprendizado de máquina, mais simples será para o usuário entender a decisão dada pelo modelo.

Ainda de acordo com Molnar (2018), em alguns casos, o motivo da decisão tomada é irrelevante; importa apenas saber se o software é preciso na sua decisão. Contudo, acontecimentos inesperados deixam o ser humano curioso. Por exemplo, o processo através do qual determinado site de vendas chegou à conclusão de que o usuário se interessa por aquele(s) produto(s) específico(s). Na Figura 1 é possível perceber um tipo de propaganda de um produto que aparece para o usuário.

Figura 1 - Propaganda de site de vendas

Fonte: adaptado de Magazine Luiza (2020).

Para Molnar (2018), quanto mais a decisão de uma máquina influencia a vida de um usuário, mais importante é explicar o motivo daquela decisão. Entretanto, segundo Adadi e Berrada (2018), os algoritmos de inteligência artificial sofrem de opacidade: é difícil descobrir o caminho que o software percorreu para chegar aquela decisão, ou é raro que um deles exponha o motivo.

Nos últimos anos, muitas pesquisas envolvendo sistemas de recomendação foram feitas, com o objetivo de tornar as recomendações cada vez mais precisas, aprimorando os algoritmos que realizam a recomendação (SHANI e GUNAWARDANA, 2011).

Porém, na maioria dos casos, os desenvolvedores dedicam muito tempo na qualidade da recomendação e não implementam métodos com o objetivo de explicar ao usuário o porque de um item específico está sendo recomendado a ele. Algumas vezes recomendações de objetos são feitas, sem que o utilizador entenda o motivo, levando o usuário a ignorar a recomendação ou até mesmo questionar o porque daquele item específico está sendo recomendado a ele.

Ainda segundo Molnar (2018) muitas vezes, saber a razão pela qual uma decisão foi tomada pode agregar conhecimento ao usuário que está acatando a decisão. A opacidade, citada por Adadi e Berrada (2018) em alguns casos atrapalha o avanço do conhecimento em algumas áreas. Além disso, em alguns casos, a interpretabilidade não é só importante, mas também essencial. Suponha que um sistema de recomendação desenvolvido para auxiliar profissionais da área de medicina na recomendação de tratamentos para câncer de mama, por exemplo, recomende um determinado método terapêutico para algum paciente. É importante

para o profissional saber o motivo daquela recomendação, não só para agregar conhecimento a ele, mas também para analisar se o tratamento é realmente eficaz.

1.2. Objetivo geral

O objetivo geral da pesquisa é desenvolver um sistema de recomendação, aplicando a técnica de filtragem colaborativa para se gerar as recomendações. A técnica escolhida é amplamente utilizada por pesquisadores e profissionais, tendo em vista sua praticidade e bons resultados. Objetiva-se também gerar um grafo que relacione o gênero musical e, por meio dele, justificar ao usuário o porque da recomendação dada.

1.3. Objetivos específicos

Para tanto, alguns objetivos específicos são necessários. Uma das primeiras etapas para se chegar ao objetivo geral do trabalho é o desenvolvimento do software de recomendação, utilizando a base de dados sobre informações musicais. Após o software estar finalizado, é necessário gerar as recomendações para cada usuário.

Com a posse das recomendações, é necessário montar o grafo que relacione os gêneros musicais. Para isso, é preciso extrair informações estruturas da Wikipédia sobre cantores, bandas e gêneros musicais.

O grafo irá utilizar principalmente os dados sobre gêneros musicais e, com isso, será possível determinar subgêneros de gêneros, a origem estilística do gênero e os instrumentos que são geralmente utilizados nos gêneros.

Tendo a posse das recomendações e do grafo, é possível determinar o gênero dos artistas ou bandas recomendadas e com isso, relacionar o que foi recomendado com o que foi ouvido pelo usuário.

1.4. Justificativas

Com a popularização da internet, juntamente com a praticidade da sua utilização no que diz respeito a quantidade de aparelhos que hoje conseguem acessá-la, a quantidade de informações geradas por esses aparelhos aumentou exponencialmente. Juntamente com isso, dados sobre os usuários são gerados

frequentemente, o que possibilita a recomendação de coisas específicas para o perfil de cada usuário individualmente.

Utilizando como exemplo a Netflix[1], é possível observar o quanto o sistema de recomendação da empresa é específico. No processo de recomendação, o algoritmo analisa a interação do usuário com o serviço: o que já foi assistido e qual a nota dada pelo usuário. O histórico de outros utilizadores do streming com características semelhantes também é analisado e, ao final, as informações são cruzadas na tentativa de realizar uma recomendação mais precisa possível (NETFLIX, 2019). Além disso, o serviço oferece uma curta explicação do motivo de algumas recomendações (ver Figura 2).

Figura 2 - Recomendações Netflix

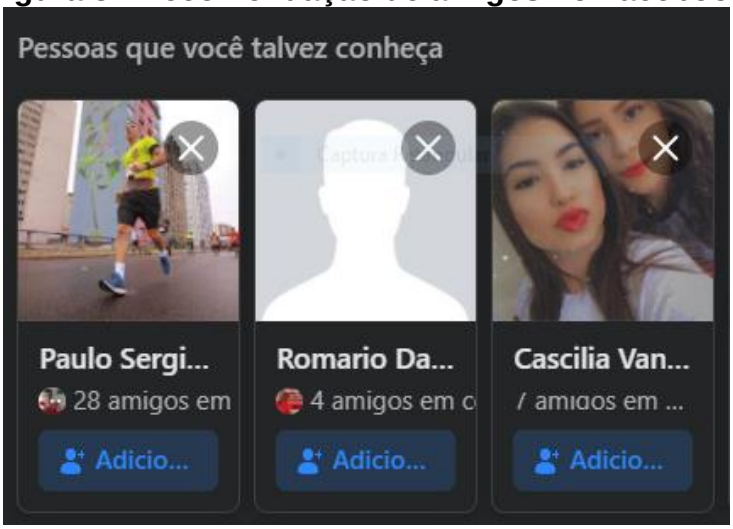
Fonte: Próprio autor.

Já na rede social Facebook[2], dentre os sistemas de recomendação presentes na plataforma, pode-se citar o algoritmo que recomenda amigos. Segundo Canaltech (2017), o algoritmo solicita acesso a informações como lista de contatos, além de acessar outros dados como check-in, comentários em publicações, amigos em comum, entre outros e com esses dados, recomenda amigos a usuários, como mostra a Figura 3.

Nos casos de streming de vídeo e música, grandes quantidades de informações são apresentadas ao usuário. A pouca experiência pessoal por parte do indivíduo que está utilizando algum desses serviços pode levá-lo a um impasse na hora da escolha do serviço específico (CAZELLA; NUNES; REATEGUI, 2010).

[1] https://www.netflix.com/
[2] https://www.facebook.com/

Figura 3 - Recomendação de amigos no Facebook



Fonte: Adaptado de Facebook (2020).

1.5. Estrutura do trabalho

O conteúdo presente neste trabalho divide-se da seguinte forma: o capítulo de revisão bibliográfica aborda assuntos essenciais para o entendimento do que foi desenvolvido, tais como World Wide Web, Resource Description Framework, Linked Data, Linked Open Data e DBpedia. Visto que o conteúdo sobre sistemas de recomendação é mais abrangente e importante, além de ser o foco desta pesquisa, o capítulo três engloba esta área, bem como um método específico de sistemas de recomendação, que é o de filtragem colaborativa, além da questão da interpretabilidade em sistemas de recomendação. Ao que se segue, o capítulo quatro traz informações sobre os meios utilizados para se alcançar o objetivo desta pesquisa. São informados os dados que foram coletados da last.fm e DBpedia, bem como o tratamento dos dados coletados da DBpedia, além do processo de geração das recomendações e do grafo. O capítulo quatro também aborda um esquema de dados desenvolvido para esta pesquisa. Por sua vez, o capítulo cinco aborda os resultados obtidos nesta pesquisa, bem como exemplos de dados gerados pelo algoritmo de justificativas. Por fim, o capítulo de conclusão traz as considerações finais sobre o trabalho, bem como as contribuições e algumas sugestões de trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

Este capítulo aborda temas relacionados ao trabalho. Para um melhor entendimento, expõe-se a visão de diversos autores, com o objetivo de embasar o estudo, dando sentido a termos citados.

Os assuntos são essenciais para o entendimento, tendo em vista que o trabalho explana sobre dados abertos vinculados e sua relação com as bases de conhecimento, bem como a importância desses dados para o desenvolvimento de sistemas de recomendação.

Os temas abordados são World Wide Web, Resource Description Framework, Linked Data, Linked Open Data e DBpedia.

2.1. World Wide Web (WWW)

A World Wide Web (WWW) reinventou os modos de comunicação mundial entre pessoas, tendo em vista as formas de interação da época, como a escrita, a fala ou videoconferência por meio de computadores interligados. No ano de 1945 Vannevar Bush, engenheiro norte-americano, publicou seu artigo intitulado "*As We May Think?*" ou, em tradução livre "Como podemos pensar?", onde ele anunciou a criação do Memex - Memory Extension: máquina com o objetivo de armazenar conhecimento, tendo em vista a quantidade de informação gerada, sem a possibilidade de armazenamento e acesso (VIEIRA, 2014). Sequencialmente, a IBM lança o Standard Generalized Markup Language ou SGML, tecnologia que permite o processamento de informações em linguagens como HTML. Em 1960, Ted Nelson já tinha o objetivo de criar uma rede mundial de computadores conectados com uma interface simples para seus usuários e acabou lançando o projeto Xanadu. Com isso, Tim Berners-Lee concluiu a WWW no ano de 1989 no Centre Européen de Recherche Nucléaire ou CERN e utilizou pela primeira vez em março de 1991; em maio de 1991, foi totalmente operacionalizada em rede e em agosto do mesmo ano, foi disponibilizada para o público (WHITEHEAD, 2007 apud VIEIRA, 2014).

Para um melhor entendimento da busca e recuperação de informações na Web é preciso entender alguns conceitos antecipadamente: multimídia, hipermídia, hipertexto e ciberespaço. Multimídia pode ser explicada como a união controlada por um computador ou dispositivo móvel de no mínimo um tipo de mídia estática, como

texto, ou uma fotografia com no mínimo um tipo de mídia dinâmica, como vídeo ou áudio, por exemplo, envolvendo o tratamento da informação digital (RIBEIRO e GOUVEIA, 2004 apud VIEIRA, 2014).

Hipertexto é definido como um sistema computacional que tem como objetivo apresentar informações em forma de texto, ligado por palavras. O leitor tem a opção de ler o texto de forma não linear: pode ler um trecho do meio ou do fim, não sendo necessária a leitura sequencial, começando do início. Hipermídia seria a interseção do conceito de multimídia com hipertexto, onde textos, vídeos, sons ou imagens seriam intercalados (REZENDE e BARROS, 2001 apud VIEIRA, 2014).

Por fim, VIEIRA (2014) define ciberespaço como a “virtualização da realidade por meio da tecnologia”. Ou seja, na medida do possível, seria criar versões virtuais de algo concreto, que existe na realidade.

Com a invenção da World Wide Web, foi possível criar um espaço mundial de informações compartilhadas (HEATH e BIZER, 2011), diminuindo as barreiras para publicação e compartilhamento de informações a nível global (BIZER; HEATH; BERNERS-LEE, 2011). Segundo Heath e Bizer (2011), a humanidade está vivendo em um mundo rodeado de dados, seja sobre o desempenho das escolas, quão eficiente é o consumo do carro, variados tipos de produtos de diferentes marcas ou dados sobre como os impostos são utilizados.

Porém, apesar da grande quantidade de benefícios que a Web trouxe com a sua criação, até recentemente os mesmos princípios utilizados no compartilhamento de informações em forma de documento não foram aplicados ao compartilhamento de dados. (BIZER; HEATH; BERNERS-LEE, 2011). Contudo, a Web está sendo cada vez mais compreendida como um espaço de compartilhamento e conexão de não apenas documentos interligados, mas também dados vinculados (BIZER et al., 2008). Nos últimos anos, a Web deixou de ser apenas um espaço de documentos vinculados e passou a ser um local de documentos e dados conectados. Segundo Bizer, Heath, Berners-Lee (2011) juntamente com essa mudança criou-se um grupo de práticas recomendadas para publicação e conexão de dados na web, que ficou conhecido como Linked Data.

2.2. Resource Description Framework (RDF)

Recomendado pela W3C, o Resource Description Framework (RDF) define uma linguagem genérica para descrever recursos na Web. Basicamente, RDF pode ser descrito como uma estrutura com o objetivo de representar e padronizar informações na Web. A estrutura básica de qualquer expressão RDF é sustentada por triplas, onde cada uma é formada por um sujeito, predicado e um objeto. (CORBY; DIENG; HÉBERT, 2000; MCBRIDE, 2004; W3C, 2004).

Para um melhor entendimento, observe a Figura 4. A tripla RDF é representada como um grafo onde o predicado, representado por uma aresta do grafo, simboliza a conexão existente entre o sujeito e o objeto, representados por nós.

Figura 4 - Tríade RDF

Fonte: Adaptado de W3C (2004).

2.3. Linked Data

Segundo Bizer, Heath, Berners-Lee (2011), Linked Data faz referência a um grupo de práticas que devem ser utilizadas como recomendação para publicação de dados vinculados na Web. Essas práticas estão sendo cada vez mais utilizadas por provedores de dados. Para Bizer et al. (2008), Linked Data significa a utilização de RDF (Resource Description Framework) e HTTP (Hypertext Transfer Protocol) para publicação de dados estruturados na Web, com o objetivo de conectar essas informações a diferentes fontes de dados. Os primeiros esboços sobre Linked Data foram propostos por Tim Berners Lee no ano de 2006, que forneceu várias orientações sobre quais foram os editores de dados que começaram a aplicar os conceitos de Linked Data (BIZER et al., 2008; W3, 2006).

Para Bizer, Cyganiak, Heath (2007), existem dois princípios básicos que regem os dados vinculados na Web, que são:

2.1.1. Usar o modelo de dados RDF quando for publicar dados estruturados na Web;

2.1.2. Usar links RDF quando for conectar dados de diferentes fontes.

A utilização de ambos os princípios citados conduz a criação de um grupo de dados na internet, onde se cria um espaço no qual pessoas ou entidades podem compartilhar e utilizar dados sobre qualquer coisa. Esse conjunto de dados que foi formado pode ser chamado de Web of Data ou Semantic Web (BIZER; CYGANIAK; HEATH, 2007).

Tim Berners Lee, no ano de 2001, também criou o conceito de Web Semântica (Semantic Web), visando a possibilidade de estruturar dados na Web (DZIEKANIAK; KIRINUS, 2004). É uma tecnologia que foi criada para compartilhar dados, bem como a Web de hipertexto serve para compartilhar documentos (BERNERS-LEE et al., 2006). Porém, Tim afirma que Web semântica não está relacionada apenas a introduzir dados na Web, mas trata-se também de vincular esses dados a outros, possibilitando a exploração desses dados por máquinas ou pessoas, de forma que ao encontrar um dado na Web, a entidade que o acessa possa encontrar vários outros dados relacionados (W3, 2006).

2.4. Linked Open Data

Para Bizer, Heath, Berners-Lee (2011), o maior exemplo de utilização dos princípios de Linked Data tem sido o projeto Linked Open Data (LOD). Resultante de um esforço da comunidade de base, o projeto foi fundado em fevereiro de 2007 e é apoiado pela W3C[3] (BIZER et al., 2008). Considerado por muitos o principal responsável pelo surgimento da Web Semântica, segundo Jain et al. (2010) o esforço investido pela comunidade Linked Open Data é essencial para a realização da visão da Web Semântica. O propósito do projeto é localizar grupos de dados disponíveis sob licença aberta na Web, publicar os dados novamente utilizando o modelo RDF e conectá-los entre si (BIZER et al., 2008).

LOD é, basicamente, Linked Data mas, nesse caso, os dados estão sob uma licença aberta, de forma que qualquer pessoa possa acessar, sem nem um tipo de restrição a grupos fechados ou organizações. Considerando que o princípio de

[3] Principal organização responsável por padronizar a criação de interpretação de conteúdo para a Wes. Mais informações em: https://www.w3.org/

Linked Data seria os dados estarem interconectados, não é necessário que os dados estejam também abertos ao acesso do público. Existe um alto uso de dados importantes, apesar de estarem ligados internamente, restritos apenas a um grupo fechado ou a alguma organização particular ou pública. (W3, 2006).

2.5. DBpedia

Segundo Bizer et al. (2009), bases de conhecimento estão exercendo uma função trivial para aperfeiçoar a inteligência de pesquisa na Web, além de auxiliar também na integração de dados. Uma grande parte das bases de conhecimento são criadas por pequenos grupos de engenheiros, dificultando a manutenibilidade ao passo que os domínios mudam. Em contrapartida, a Wikipédia tornou-se uma das principais fontes de conhecimento mundial, sendo mantida por milhões de colaboradores em todo o mundo.

Segundo o site Alexa[4], a Wikipédia é o quinto site mais acessado do mundo. A Wikipédia possui edições oficiais em 287 idiomas diferentes, sendo que em alguns idiomas existem algumas centenas de artigos e em outros (como inglês, por exemplo) possui mais de 5 milhões de artigos publicados (LEHMANN et al., 2015; WIKIPÉDIA, 2019). A enciclopédia não se limita apenas a texto livre. Nos artigos o leitor pode encontrar vários tipos de dados, infoboxes, tabelas, listas e dados de categorização. Apesar da quantidade de dados, a Wikipédia não disponibiliza recursos de pesquisa avançada, limitando-se apenas a pesquisa em texto livre e assim, seria mais difícil realizar uma consulta mais específica nos dados do site (LEHMANN et al., 2015; AUER et al., 2007). Por exemplo, segundo Lehmann et al. (2015) usando a pesquisa que a Wikipédia fornece ao usuário, seria difícil selecionar todos os rios que escoam para o rio Reno que tem mais de 100 milhas.

Outro exemplo interessante é descobrir os subgêneros de alguns gêneros musicais. A pesquisa seria extensa, tendo em vista os modos de pesquisa que a Wikipédia fornece ao usuário. seria necessário realizar a pesquisa de cada gênero, procurar no documento de texto seus respectivos subgêneros e documentar cada informação coletada.

[4] Organização que fornece dados sobre o tráfego de internet. Pertence a Amazon e pode ser acessada pelo seguinte link: https://www.alexa.com/

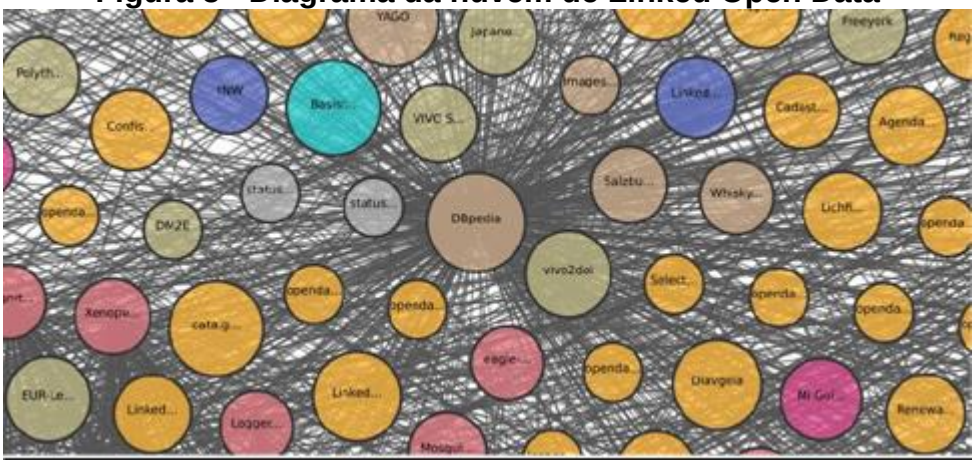
Resultante de um esforço da comunidade de crowdsourcing, o projeto DBpedia utiliza essa grande base de conhecimento e monta uma base de conhecimento multilingue por meio da extração de dados estruturados da Wikipédia em 125 idiomas diferentes. A base de conhecimento resultante descreve 4,58 milhões de entidades onde 4.22 milhões são ontologias consistentes, que incluem 1.445,000 pessoas, 735.000 lugares, 411.000 obras criativas (onde 123.000 são álbuns musicais, 87.000 filmes e 19.000 video games), 241.000 organizações, 251.000 espécies e 6.000 doenças. A DBpedia abrange os mais variados tópicos e define uma grande quantidade de links RDF para fontes externas de dados e, por isso, vários editores do Linked Data decidiram criar links RDF apontando para a DBpedia. O resultado disso é que o projeto DBpedia tornou-se um fator chave para a ascensão da iniciativa LOD, tornando a DBpedia um hub de interconexão central na Web of Data (BIZER et al., 2009; LEHMANN et al., 2015; DBPEDIA, 2019).

Em seu trabalho, Auer et al. (2007) diz que um dos grandes desafios da ciência da computação é reunir informações e conhecimentos estruturados ao redor do mundo, com o objetivo de responder consultas ricas semanticamente. Auer et al. (2007) ainda diz que se esse objetivo for alcançado, impactará diretamente no mundo como um todo, referindo-se à possibilidade de criação de software com grandes potenciais, já que estariam trabalhando com uma grande quantidade de dados, o que diminuiria as chances de erros.

Na Figura 5, é possível observar um diagrama de uma nuvem de Linked Open Data. No diagrama é possível ver o quanto a DBpedia é importante no mundo dos dados conectados. Nota-se também o quão central ela está na nuvem de LOD, sendo que a maioria das fontes de dados (senão todas) possuem links apontando para DBpedia.

Abordados os temas essenciais para o entendimento do resultado desta pesquisa, o capítulo que se segue irá servir de aprofundamento sobre o tema de sistemas de recomendação, visto que este tema é a base para a construção do que foi proposto.

Figura 5 - Diagrama da nuvem de Linked Open Data



Fonte: Adaptada de (LOD-CLOUD, 2019).

3. SISTEMAS DE RECOMENDAÇÃO

Tendo em vista a grande quantidade de dados disponível na Web atualmente, muitas pessoas encontram informações diversificadas e por vezes os indivíduos não possuem tanta experiência pessoal para conseguir escolher corretamente o que buscam em meio a quantidade de alternativas como, por exemplo, o número exorbitante de vídeos disponíveis para entretenimento, bem como músicas, filmes ou até mesmo produtos do e-commerce.

Analogamente ao mundo real, onde confiamos nas recomendações de outras pessoas, os sistemas de recomendação ajudam no processo de sugestão, baseado nas relações interpessoais. Um sistema de recomendação, como mostra a Figura 6, pode ser caracterizado como um software com o objetivo de ajudar usuários a encontrar conteúdos de interesse dentre um conjunto de opções e assim, facilitar a busca por informações relevantes para o utilizador (CAZELLA e REATEGUI, 2005; CAZELLA; NUNES; REATEGUI, 2010).

Figura 6 - Sistema de recomendação

Fonte: Adaptado de IBM (2014).

Apesar de o objetivo ser o mesmo, que é a recomendação, para Vieira e Nunes (2012), há três tipos de sistemas de recomendação que utilizam diferentes formas de filtragem de conteúdo: filtragem colaborativa, baseada em conteúdo e híbridos.

Um sistema de recomendação que utiliza filtragem colaborativa funciona por meio da comparação de informações de usuários e a semelhança entre eles. Dessa forma, os objetos serão recomendados a um determinado usuário ou grupo de usuários porque anteriormente utilizadores com características semelhantes a ele já utilizaram os mesmos objetos (VIEIRA e NUNES, 2012). O sistema Tapestry (GOLDBERG et al., 1992) é considerado o primeiro sistema de recomendação. Seus desenvolvedores foram os primeiros a utilizar o nome filtragem colaborativa para designar o método de filtragem do sistema (CAZELLA; NUNES; REATEGUI, 2010; GOLDBERG et al., 1992).

Sistemas de recomendação baseados em conteúdo realizam sua filtragem com base nos itens similares que o usuário já utilizou. O método baseado em conteúdo recupera informações explícitas de metadados, ou implícitas, onde se compara os dados recuperados e sua similaridade com o que será filtrado. Contudo, uma grande parte dos sistemas de recomendação utilizam as duas técnicas, onde se aproveita a filtragem colaborativa de acordo com informações de usuários semelhantes e objetos já utilizados pelos usuários. A esses sistemas de recomendação, dá-se o nome de híbridos (VIEIRA e NUNES, 2012).

3.1. Filtragem colaborativa

Segundo Schafer (2001, apud CAZELLA et al., 2009), a filtragem colaborativa consolidou-se com uma das técnicas mais utilizadas por sistemas de recomendação. Segundo Cazella et al. (2009), a filtragem colaborativa tem como fundamento o compartilhamento de informações entre pessoas a respeito de suas experiências sobre utilização de serviços de streaming, compra de produtos, entre outros.

Algoritmos de filtragem colaborativa são amplamente utilizados devido sua fácil implementação, simplicidade e praticidade (LIAO e LEE, 2016) e por isso, vem se destacando dentre os algoritmos de recomendação (COSTA, 2014). De acordo com SANTOS (2017), conforme citado por RICCI et al. (2011), a filtragem colaborativa se baseia na ideia de que existe a possibilidade de um usuário ter interesse por itens com boas avaliações dadas por usuários com perfis similares.

O sistema de recomendação baseado em filtragem colaborativa tem como objetivo prever a preferência de usuários, a variar de acordo com o nicho no qual o sistema de recomendação está atuando.

De acordo com Cazella et al. (2009), a melhor forma de se recomendar itens seria com base em avaliações dadas por usuários sobre os itens. Com isso, é possível observar o comportamento de um grupo de usuários qualquer e, por meio disso, verificar a similaridade de preferências dentro do grupo. Então, tendo a posse de informações de cada usuário, é possível buscar os seus “vizinhos mais próximos”, classificados de acordo com a similaridade de comportamento entre si (ADOMAVICIUS e TUZHILIN, 2005).

O funcionamento de um sistema de recomendação parte dos serviços de entretenimento até áreas mais vitais, como a saúde. Imperceptíveis a alguns usuários, eles estão presentes no dia-a-dia de todos que utilizam smartphones, computadores e outros dispositivos que se conectam a internet.

Porém, Segundo SANTOS (2017), apesar das vantagens de se utilizar a filtragem colaborativa para recomendar filmes, músicas, produtos e etc, essa abordagem nem sempre traz bons resultados. Alguns sites, como Netflix, Amazon e Spotify possuem coleções de dados dispersos, o que dificulta a aplicação de técnicas de filtragem colaborativa para se realizar recomendações, tendo em vista a quantidade de tempo gasto para se realizar a recomendação (PARK et al., 2015).

Segundo SANTOS (2017), a filtragem colaborativa com agrupamento de usuários consegue lidar com o problema da dispersão dos dados, sem afetar o desempenho do sistema.

O método *k-Nearest Neighbors,* kNN ou k vizinhos mais próximos (em tradução livre) gera as recomendações a partir das informações encontradas nos *k* vizinhos mais próximos ao usuário que está sendo alvo da recomendação (COSTA, 2014).

Para um melhor entendimento, Adomavicius e Tuzhilin (2005) explicam o funcionamento da filtragem colaborativa da seguinte forma: suponha que um aplicativo de recomendação de filmes, com intuito de recomendar algum filme para o usuário *y*, comece a vasculhar os vizinhos mais próximos do usuário em questão. Para isso, é necessário analisar quais filmes o usuário *y* assistiu e classificou como bom e verificar se o usuário *x* também classificou o filme como bom. Por meio disto, o algoritmo deduz que se *x* gostou do mesmo filme que *y*, provavelmente os filmes assistidos por *x* serão classificados como bons por *y*, que ainda não assistiu. Com isso, o algoritmo de recomendação irá recomendar para o usuário *y* os filmes que o usuário *x* assistiu, mas que ainda não foram assistidos por *y*. Observe a Figura 7.

Figura 7 - Lógica abstrata do funcionamento de um sistema de recomendação baseado em filtragem colaborativa

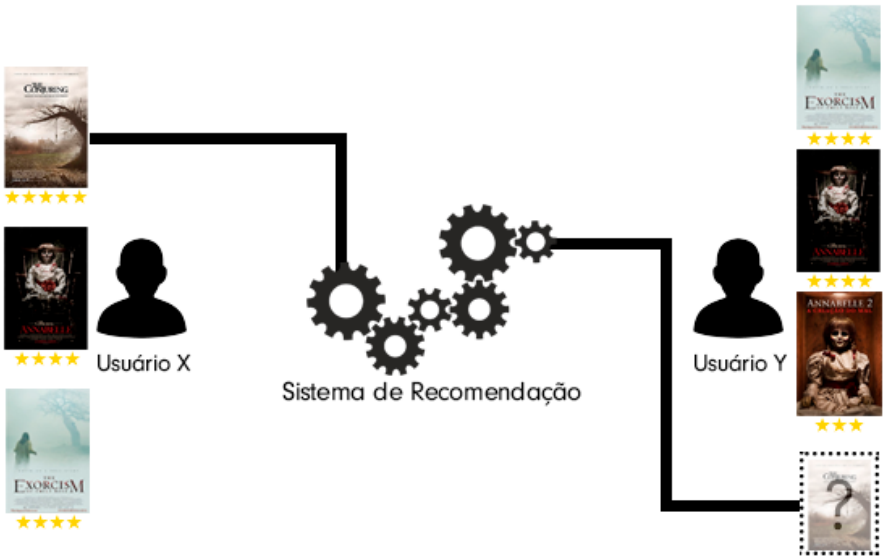
Fonte: Próprio autor.

Na figura acima é possível observar que o usuário X classificou os filmes *"The Conjuring", "Annabelle"* e *"The exorcism of Emily Rose".* Para esse exemplo, as classificações são em forma de estrela, onde a pontuação é de 1 a 5 estrelas. O usuário X classificou os filmes respectivamente com cinco, quatro e quatro estrelas. Observa-se também que o usuário Y classificou os filmes *"Annabelle: Creation", "Annabelle"* e *"The exorcism of Emily Rose".* A classificação dada pelo usuário Y foi respectivamente três, quatro e quatro estrelas. Ainda sobre a Figura 7, o sistema de recomendação, ao precisar recomendar um filme para o usuário Y, coleta as informações de ambos os usuários. Com isso, o sistema percebe que tanto o usuário X quanto o usuário Y assistiram *"Annabelle"* e *"The exorcism of Emily Rose"* e os classificaram com quatro estrelas. Ao analisar o perfil do usuário Y, o sistema percebe que ele não assistiu o filme *"The Conjuring"*, porém X assistiu e o classificou com cinco estrelas. Com isso, o sistema decide recomendar *"The Conjuring"* ao usuário Y, tendo em vista a semelhança de preferências dos usuários. Observe a Figura 8.

Os exemplos descritos relacionam apenas um usuário a outro. Porém, quando o sistema de recomendação está atuando, os perfis de diversos usuários são analisados levando em consideração suas preferências.

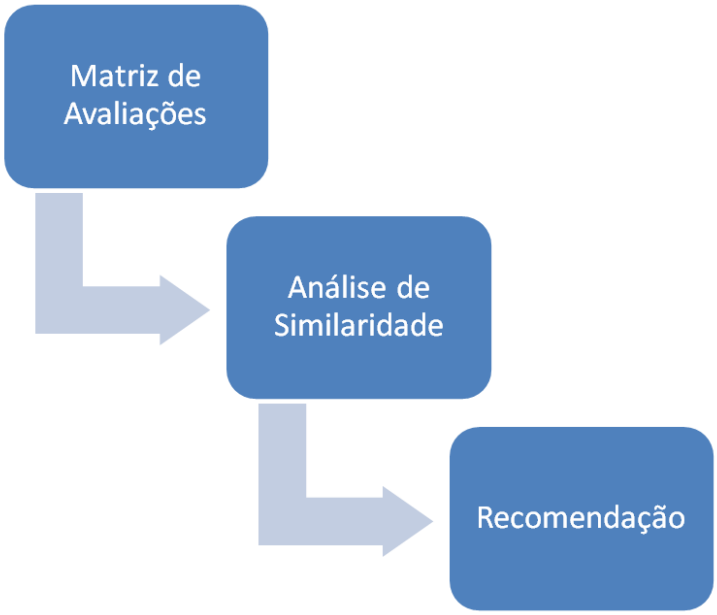
Para um melhor entendimento do funcionamento de um sistema de recomendação, observe a Figura 9 a seguir.

Figura 8 - Geração das recomendações



Fonte: próprio autor.

Figura 9 - Fluxo de atividades de um sistema de recomendação



Fonte: Próprio autor.

O primeiro processo ilustrado na imagem é o desenvolvimento de uma matriz *n* x *m*, onde *n* seria a quantidade de usuários presentes em uma base de dados e *m* a quantidade de itens que podem ou não estarem classificados pelos usuários. Suponha que a matriz exemplificada seja de classificação de músicas e que as informações presentes nela seriam usuários que classificaram músicas ouvidas em um sistema de classificação de um a cinco. É preciso levar em consideração que a matriz em questão pode ter variados tipos de dados. Por exemplo, para um site de

vendas, a matriz pode trazer a informação se o usuário comprou ou não determinado produto. Para uma rede social, a informação pode ser quanto tempo o usuário ficou em uma publicação.

Após a finalização da matriz de classificação, suponha que é necessário que o sistema tente prever a recomendação de um usuário u_i, onde i = 1, 2, 3, …, n, para o item i_j, onde j = 1, 2, 3, …, m. O usuário u_i, claro, ainda não classificou o item i_j. O próximo passo será a análise de similaridade entre usuários. Por sua vez, para o exemplo em questão, o sistema de recomendação irá calcular para cada usuário o resultado de similaridade, utilizando a função de similaridade. Existem diversas funções para o cálculo de similaridade entre usuários. As suas aplicações variam de acordo com o tipo de informação contido na matriz. Observe a função a seguir.

$$\frac{\sum i \in I \left(r_{x,i} - \overline{r_x}\right)\left(r_{y,i} - \overline{r_y}\right)}{\sqrt{\sum i \in I \left(r_{x,i} - \overline{r_x}\right)^2 \sum i \in I \left(r_{y,i} - \overline{r_y}\right)^2}}$$

Descrevendo os componentes da função, $r_{x,i}$ seria o usuário que o sistema pretende prever a classificação para o item i_j (ou seja, o usuário u_i), bem como sua classificação para algum item que outro usuário de sua semelhança também classificou. $\overline{r_x}$ é a média de classificações que o usuário u_i dá aos itens. $r_{y,i}$ é a classificação de outro usuário que também classificou a mesma música que o usuário u_i. $\overline{r_y}$ é a média de classificações dada pelo usuário semelhante a u_i. O resultado para a equação varia entre 0 e 1, onde quanto mais próximo de 1 for o resultado, mais semelhante os usuários são.

Com o resultado da função de similaridade, o sistema analisa quais usuários possuem maior similaridade com o usuário u_i. Após a classificação, é necessário observar se os usuários similares classificaram o item i_j. Em posse dessas informações, o próximo cálculo a ser realizado é o de predição. Observe a função a seguir.

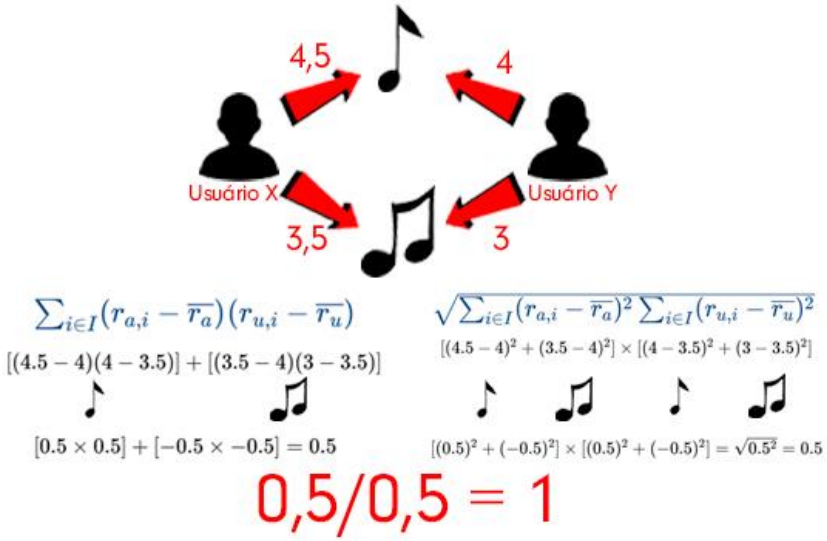
$$p_{x,i} = \overline{r_x} + \frac{\left(r_{y,i} - \overline{r_y}\right) x W_{x,y}}{\sum y \in k \, w_{x,y}}$$

Comparada a função de similaridade, a função de predição possui apenas algumas informações diferentes. $\overline{r_x}$ continua sendo a média de notas dada pelo usuário u_i, bem como $\overline{r_y}$ é a média de notas dada pelo usuário semelhante. $r_{y,i}$ nesse

caso é a nota dada pelo usuário semelhante ao item i_j. Essa informação é útil tendo em vista que o sistema pretende prever a nota que o usuário u_i pretende dar ao item i_j. Por fim, $w_{x,y}$ é o resultado da função de similaridade calculada anteriormente. O resultado da fórmula de predição é a provável nota que o usuário u_i daria ao item i_j.

Suponha que, para o exemplo que se segue, o sistema pretende prever a classificação de um item ainda não classificado por um usuário X. O sistema tem a nota do usuário X para outros itens, bem como a de outro usuário, chamado Y. Com isso, o sistema calcula a nota de similaridade. Observe a Figura 10.

Figura 10 - Resultado da função de similaridade



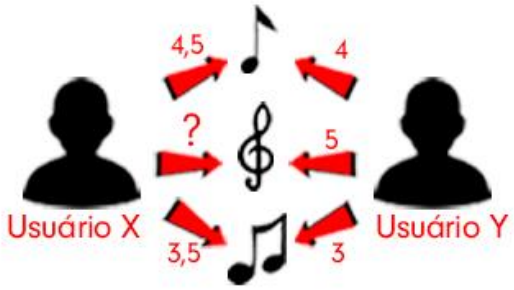
Fonte: próprio autor.

O resultado da função de similaridade entre os dois usuários deu igual a 1. O sistema, ciente que o usuário Y classificou o item que se pretende prever a nota de X e com a posse do resultado da função de similaridade entre os dois usuários, realiza o cálculo de predição. Observe a Figura 11.

Para um melhor entendimento do funcionamento de sistemas de recomendação com filtragem colaborativa, observe a Figura 12 que traz o pseudocódigo de seu funcionamento.

Os parâmetros da função de filtragem podem ser descritos da seguinte forma: *itensParaClassificacao* é a lista de itens que o sistema pretende prever. *itensClassificados* são os itens que o usuário já classificou. *u* são os vizinhos mais próximos do usuário e *k* a quantidade de vizinhos que serão considerados para a previsão.

Figura 11 - Resultado do cálculo de predição



$$4 + \frac{5-3.5\times1}{1} = 5.5$$

Fonte: próprio autor.

Figura 12 - Pseudocódigo de algoritmo de filtragem colaborativa

```
programa
{

    funcao colaborativeFilteringKnn (itensParaClassificacao[], itensClassificados[], u[], k)
    {
        inteiro item, classificados
        item = itensParaClassificacao[].tamanho
        classificados = itensClassificados[].tamanho
        enquanto (item > 0) faca
        {
            enquanto (classificados > 0) faca
            {
                se (itensParaClassificacao[item] <> intesClassificados[classificado])
                {
                    itensParaClassificacao[item] = claDeAcoComViziMaisProx (k, u[], itensParaClassificacao[])
                    classificados = classificados - 1
                }
            }
            item = item - 1
        }
        classificacaoDecrescente (itensParaClassificacao[])
        retorne itensParaClassificacao[]
    }
}

```

Fonte: próprio autor

Antes de realizar a previsão, o algoritmo verifica se o usuário já classificou algum item da lista. Caso não, o algoritmo atribui a classificação para o item em questão, invocando a função de classificação, passando como parâmetros *k, u,* e *itensParaClassificacao.* Após realizar esse procedimento para todos os itens em ambas as listas (*itensClassificados* e *itensParaClassificacao*), o sistema ordena as classificações das menores para as maiores e retorna a lista *itensParaClassificacao* novamente, com a previsão de classificação.

3.2. Interpretabilidade sobre decisões

Segundo Carvalho et al. (2019), não há como não notar a presença dos sistemas de aprendizado de máquina no dia-a-dia na sociedade. Cada vez mais eles estão presentes, nas mais diversificadas áreas. Além disso, sua expansão é inquestionável, tendo em vista a quantidade de pesquisas que cada vez mais ampliam os horizontes da área de aprendizado de máquina. Isto posto, é visto que as decisões tomadas por esses algoritmos repercutem cada vez mais no dia-a-dia dos usuários enquanto sociedade. Porém, apesar de suas crescentes utilizações juntamente com seus impactos, uma grande parte desses sistemas de aprendizado de máquina continuam opacos para os usuários. Contudo, ainda segundo Carvalho et al. (2019), novas normas tornaram obrigatórios audições de autenticidades para as decisões tomadas, o que aumenta a possibilidade de confiar em sistemas de decisões.

Para Miller (2018, apud MOLNAR, 2018), interpretabilidade pode ser definida como o grau que o ser humano é capaz de entender uma decisão tomada. Para Doshi-Velez e Kim (2017), a definição de interpretabilidade se assemelha a dada por Miller: a interpretabilidade é definida como a capacidade de justificar em termos compreensíveis a um ser humano.

Porém, para Molnar (2018), nem sempre é necessário que um software justifique sua decisão. Em alguns casos, o desenvolvedor deve escolher entre implementar ou não métodos que justificam a decisão, pois nem sempre é interessante implementar métodos que aumentem a interpretabilidade do sistema, tendo em vista que em alguns casos só é interessante para o usuário saber o resultado da decisão e a utilização de métodos de interpretabilidade refletem no desempenho do sistema, na hora da tomada de decisão. Carvalho et al. (2019) complementa, afirmando que em muitos casos o alto desempenho diante das predições dos sistemas de aprendizado de máquina são suficientes, descartando a necessidade da interpretabilidade. Doshi-Velez e Kim (2017) definem duas situações nas quais a interpretabilidade é considerada desnecessária:

- Quando não existe a possibilidade de grandes consequências para resultados equivocados;

- O problema sobre o qual o sistema trabalha é bem estudado e já foi validado em aplicações da vida real, onde se pode confiar na decisão do sistema, apesar da sua imperfeição.

Porém, Doshi-Velez e Kim (2017) prosseguem afirmando que apenas um único parâmetro como, por exemplo, a precisão do resultado de um algoritmo de aprendizado de máquina é uma exposição incompleta de grande parte das problemáticas do mundo real. Molnar (2018) complementa afirmando que existem situações nas quais saber o porquê pode ajudar na compreensão de um problema, dos dados e do motivo pelo qual um sistema pode falhar. Em alguns casos, o resultado final não é o bastante. Mais que isso, é necessário explicar quais os passos percorridos para se chegar aquela decisão.

Doshi-Velez e Kim (2017) exemplificam algumas situações nas quais a interpretabilidade possui sua relevância:

- Para a compreensão científica: dado que não há uma definição perfeita de conhecimento, a melhor maneira de adquiri-lo é com explicações que possam ser convertidas em conhecimento;
- Para a segurança: especialmente em tarefas complexas, quase nunca é possível testar um sistema completamente. Existem possíveis entradas, bem como saídas que não estão previstas;
- Para a questão ética: a noção de justiça do ser humano é abstrata demais, o que torna inviável sua implementação em um sistema. Por exemplo, ainda que classifiquem diversos tipos de preconceitos, podem existir outros tipos que não foram considerados pelos desenvolvedores;

Para um melhor entendimento da importância da interpretabilidade Molnar (2018) continua explanando a respeito de uma curiosidade sobre os seres humanos: o ser humano possui em sua mente um modelo de ambiente. Esse modelo é alterado quando algo inesperado acontece. Por exemplo, ao comer uma pizza e adoecer do estômago posteriormente, o indivíduo percebe que sempre que ele ingere aquele tipo de alimento, sentirá dores no estômago. Com isso, o modelo mental de ambiente dele é atualizado e, portanto, ele evitará comer pizza novamente. Fazendo uma analogia ao ser humano, quando softwares de aprendizado de máquina opacos, que não são transparentes quanto a sua decisão, são utilizados em pesquisas, os resultados das descobertas científicas continuam

ocultas, já que não foi informado como o software chegou àquele resultado. Aumentar os níveis de interpretabilidade e fornecer explicações são etapas essenciais para facilitar a aprendizagem e entender a razão das previsões feitas por máquinas.

Um exemplo concreto e real da importância da interpretabilidade em sistemas de aprendizado de máquina é o problema que envolve um questionário chamado COMPAS, implementado em um sistema de aprendizado de máquina que possui o mesmo nome. A sigla COMPAS é a abreviação para o termo "*Correctional Offender Management Profiling for Alternative Sanctions"* ou "perfil de gerenciamento de infrator correcional para sanções alternativas" em tradução livre (MAYBIN, 2016).

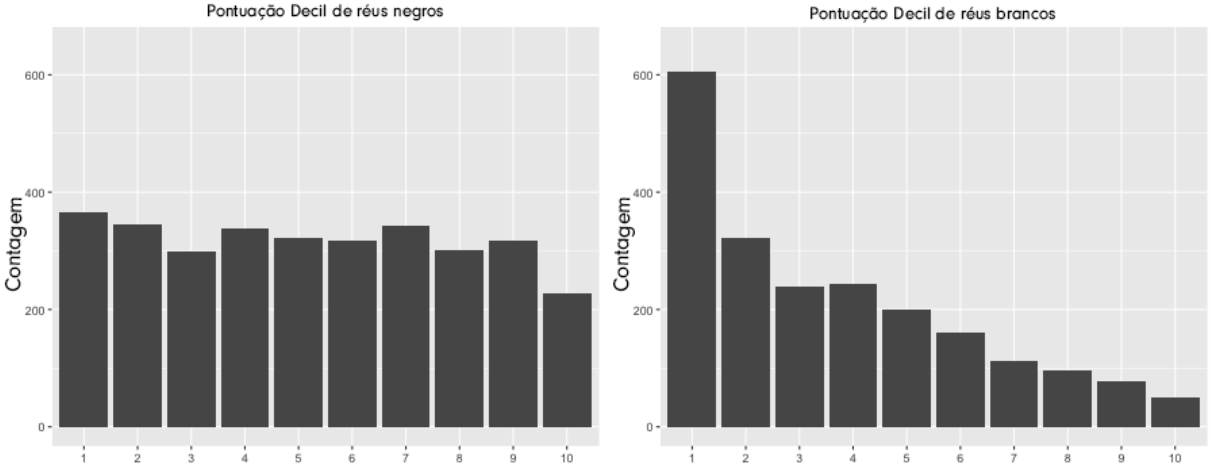
Segundo Larson et al. (2016), algoritmos de auxílio para avaliação da possibilidade de reincidência de réus estão sendo cada vez mais utilizados no Estados Unidos. Larson et al. (2016) analisou um dos algoritmos que está sendo mais utilizado, o COMPAS. O software em questão foi desenvolvido pela Northpointe, Inc e vem sendo amplamente utilizado por juízes e agentes de condicional pelo país. A análise tinha como objetivo a precisão do resultado de reincidência do algoritmo, bem como sua possível tendência contra certos grupos. A análise contou com a participação de mais de 10.000 réus no condado de Broward, Flórida.

Uma das integrantes da equipe que analisou o COMPAS, Julia Angwin, explica que a base para a decisão do sistema é um questionário, no qual o réu responde perguntas com pontuações que variam entre 1 e 10. As perguntas abordam diversas questões, como antecedentes criminais de familiares, a área em que o réu mora, entre outras questões. O resultado serve de auxílio na decisão a ser tomada quanto a forma como o réu deve cumprir sua pena: se a pessoa pode ser solta com pagamento de fiança, se deve ser presa, entre outros. O objetivo do algoritmo seria auxiliar na sentença de réus de forma mais justa, independente de erros humanos, sem preconceitos ou racismo. Porém, a equipe que analisou o COMPAS descobriu que os réus negros têm 45% a mais de chances em comparação a réus brancos de receberem uma pontuação alta no algoritmo. (MAYBIN, 2016).

Vale ressaltar que, apesar do problema relacionado ao algoritmo, o questionário não traz perguntas específicas sobre a etnia do réu. Contudo, a equipe que analisou o algoritmo informou que as questões analisadas pelo software podem

ser consideradas expressões de situações raciais como, por exemplo, os antecedentes criminais de familiares ou se o réu em questão possui antecedentes (MAYBIN, 2016). Na Figura 13 é possível observar o resultado da pesquisa, bem como a tendência do algoritmo contra réus negros.

Figura 13 - Resultado da análise do algoritmo COMPAS



Fonte: adaptado de LARSON (2016).

Apesar dos resultados do algoritmo COMPAS, segundo Maybin (2016) a empresa que desenvolveu o algoritmo se negou a esclarecer o modo como as entradas são ponderadas pelo sistema.

Tendo em vista o que foi abordado neste capítulo, o que se segue são os meios utilizados para se alcançar o objetivo geral desta pesquisa.

4. METODOLOGIA

Esta seção aborda os meios utilizados para se alcançar o objetivo geral deste trabalho. A princípio fala-se sobre os dados que foram coletados, seus tratamentos e o resultado. Logo após aborda-se a maneira que os dados são utilizados para se desenvolver o sistema de recomendação e gerar as justificativas.

4.1. Dados coletados da last.fm

Para a geração das recomendações, foram utilizadas duas bases de dados: last.fm e DBpedia. Segundo Oliveira (2020), last.fm é um tipo de rede social, na qual os usuários que a utilizam compartilham informações sobre suas preferências musicais. As músicas ouvidas pelos usuários são registradas e geram informações, como o título da música, álbum, artista, etc. Além disso, segundo Amaral e Aquino (2008) e Last.fm (2019), a plataforma conecta pessoas com interesses musicais semelhantes, recomendando músicas de acordo com o perfil do usuário. Tendo em vista que a plataforma last.fm não disponibiliza bases de dados, as informações utilizadas foram coletadas e disponibilizadas no trabalho *The lfm-1b dataset for music retrieval and recommendation,* de Schedl (2016).

Segundo Schedl (2016), sua base conta com mais de um bilhão de registro de músicas ouvidas. Esses dados foram gerados por mais de cento e vinte mil usuários que foram anonimizados na base. Cada informação coletada possui, no tangente à música, nome do artista, álbum, faixa, data e hora. Os dados coletados por Schedl foram registrados entre os períodos de janeiro de 2013 a agosto de 2014. Para a finalidade do trabalho, foi utilizada apenas uma amostragem de 997 usuários aleatoriamente selecionados, tendo em vista a quantidade abrangente de dados no arquivo. Além disso, alguns usuários não possuíam histórico e por isso, não foram acrescentados na amostra. Na Tabela 1 é possível observar os tipos de dados que foram utilizados do arquivo.

Tabela 1 - Descrição dos dados utilizados do arquivo para se gerar a recomendação

Atributo	Descrição
album-id	Código identificador do álbum musical
artist-id	Código identificador do artista
user-id	Código identificador do usuário
timestamp	Registro do evento musical contado em segundos a partir da data de 1 de janeiro de 1970
country	País de registro do evento musical
age	Idade do usuário
gender	Gênero do ouvinte
artist-name	Nome do artista do registro musical
album-name	Nome do álbum do registro musical

Fonte: Próprio autor.

4.2. Dados coletados do site DBpedia

A outra base de dados que traz informações sobre a relação entre os gêneros foi extraída do site DBpedia, que coleta dados estruturados da Wikipédia, uma enciclopédia criada de forma colaborativa. A DBpedia disponibiliza mais de uma forma de acesso a consultas de dados. Por exemplo, utilizando a linguagem SPARQL, uma linguagem que tem como objetivo consulta RDF (SEGUNDO, 2017), é possível recuperar informações estruturadas específicas, como a citada no exemplo da seção.

Não só isso, a DBpedia também disponibiliza informações em formatos de arquivos pré organizados CSV e JSON. Para o objetivo deste trabalho, os arquivos CSV's eram satisfatórios. Os arquivos recuperados do site da DBpedia foram *MusicGenre, MusicalArtist* e *Band.*

O CSV *MusicGenre* traz informações sobre os gêneros músicais de forma específica e conta com 1229 gêneros e 23 dados específicos sobre cada gênero. Já o arquivo *MusicalArtist* possui 50978 artistas, com 16 informações específicas para

cada um. Por sua vez, o arquivo *Band* traz 33613 bandas com 16 informações para cada uma.

Os dados para *MusicGenre* incluem os mais variados tipos de informações, como nome, subgênero, instrumentos, origem estilística entre outros. Da mesma forma os arquivos *MusicalArtist* e *Band* trazem informações como nome, gênero, ano de início e fim de carreira, local de criação (ou nascimento nos casos de *MusicalArtist)*, entre outros. Observe a Figura 14.

Figura 14 - Arquivos extraídos da DBpedia

Fonte: Próprio autor.

Na Tabela 2 é possível observar a quantidade de dados em cada arquivo extraído da DBpedia.

Tabela 2 - Quantidade de dados por arquivo

Arquivo	Quantidade
MusicGenre	1229
MusicalArtist	50978
Band	33613

Fonte: Próprio autor.

4.3. Tratamento dos dados coletados da DBpedia

As bases de dados retiradas da DBpedia possuíam informações que eram desnecessárias para o intuito do trabalho. De início, da base que trazia informações sobre os gêneros musicais utilizou-se apenas o gênero e o subgênero correspondente. Porém, posteriormente, foi necessário extrair também as informações sobre instrumentos que eram utilizados pelos gêneros e sua origem estilística.

Já das bases que traziam informações a respeito das bandas e cantores, as únicas informações extraídas foram o nome da banda ou do cantor e respectivamente, seu gênero. Tendo em vista que as recomendações serão dadas independentemente de se tratar de uma banda ou de um artista solo, as informações dessas duas bases serão unificadas, compondo uma base com 84591 bandas e artistas solo. Na Tabela 3 está a quantidade de dados após a união dos arquivos.

Tabela 3 - Dados unificados de bandas e artistas.

Arquivo	Quantidade
MusicalArtistBands	84591

Fonte: próprio autor.

4.4. Geração das recomendações

A linguagem utilizada para gerar a recomendação foi a linguagem R, que tem como principal objetivo a criação de gráficos estatísticos e a análise de dados. É um projeto de software livre, desenvolvido pelo *Bell Laboratories*. Uma das principais vantagens do R é a facilidade de criação de gráficos de boa qualidade (R PROJECT).

O ambiente de desenvolvimento utilizado para a geração das recomendações foi o RStudio, que tem como objetivo a produção de um ambiente de código aberto com foco na linguagem R, voltado para a ciência de dados, comunicação técnica entre colaboradores e pesquisas científicas (RSTUDIO). Para se gerar as recomendações, foi necessária a utilização de três bibliotecas: *tidyverse, data.table* e *recommenderlab.*

O pacote *tidyverse* tem a finalidade de manipular os dados. Por meio de sua utilização, é possível criar, modificar e selecionar colunas de uma base de dados, bem como filtrar linhas, ordenar e sumarizar a base, entre outros. Sua utilização foi necessária para tratar os dados antes de realizar a recomendação, tendo em vista que a base fornecida possuía informações desnecessárias para o objetivo do trabalho (WICKHAM et al., 2019).

A biblioteca *data.table* foi utilizada com o objetivo de agilizar no carregamento da base de dados. O pacote *recommenderlab* possui funcionalidades semelhantes ao pacote *data.table*, no que diz respeito ao carregamento e gravação de dados na memória. Porém, em termos de agilidade no carregamento dos dados na memória, o pacote *data.table* se mostrou superior ao *recommenderlab*, onde ao utilizar *data.tablei*, o tempo de carregamento para os dados foi em torno de 15 segundos. Já na utilização do recommenderlab, os dados levaram mais de 10 minutos para serem carregados na memória.

A utilização da biblioteca *recommenderlab* foi necessária tendo em vista sua finalidade. Diversos projetos de código aberto foram desenvolvidos com a finalidade de criar aplicações de sistema de recomendação. *recommenderlab* se difere dos demais, pois ele fornece uma infraestrutura voltada à pesquisa, desenvolvimento e teste algoritmos de recomendação. O objetivo da biblioteca é o tratamento de dados de forma consistente e eficiente, além da facilitação no desenvolvimento de algoritmos de recomendação, desenvolvimento de experimentos e análise dos resultados. Além disso, o *recommenderlab* foi desenvolvido voltado para algoritmos de recomendação baseados em filtragem colaborativa (HAHSLER, 2015), que é o objetivo deste trabalho.

Na Tabela 4 é possível observar as classes de *recommenderlab* que foram utilizadas para gerar a recomendação, bem como suas finalidades.

Além das classes utilizadas neste trabalho existem ainda diversas classes a serem exploradas pelo pacote *recommenderlab.* Para o objetivo deste trabalho, as classes citadas acima foram satisfatórias.

Inicialmente, os dados coletados do site last.fm e tratados no trabalho de Schedl (2016), são carregados no R. Devido a questões semânticas, o nome de algumas colunas da base de dados são alteradas: *artist-id* é substituido por *IdArtist* e *user-id* é substituído por *IdUser*.

Feita as alterações, para fins de melhoria na justificativa, uma base de dados é gerada com a quantidade de vezes que cada usuário ouviu cada artista.

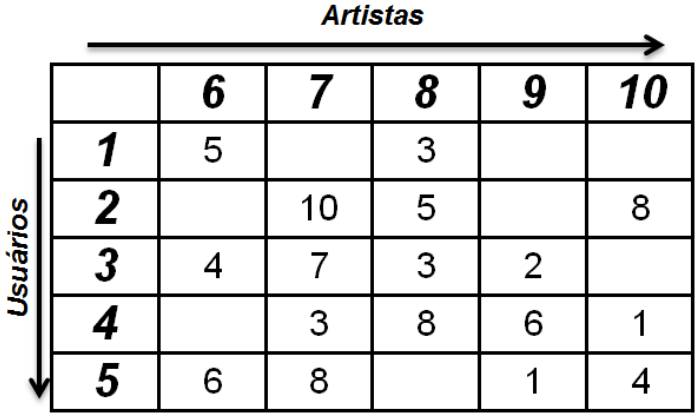
Tabela 4 - Classes utilizadas do pacote recommenderlab

Classe	Descrição
realRatingMatrix	Classe para gerar matrizes de classificação.
Recommender	Desenvolve um modelo de recomendação por meio de dados fornecidos.
predict	Gera recomendações com base em um modelo de recomendação já criado (geralmente utilizando a classe *Recommender),* juntamente com dados de novos usuários.

Fonte: próprio autor.

Posteriormente, utilizando a classe *realRatingMatrix* do pacote *recommenderlab*, gera-se uma matriz de classificação. Essa matriz possui as informações da base de dados gerada anteriormente com a quantidade de eventos de escuta de cada usuário para cada artista. Por exemplo, se o usuário 1 ouviu 5 vezes o artista 6, não ouviu o artista 7, ouviu 3 vezes o artista 8 e não ouviu o artista 9 e 10, a matriz é preenchida de acordo. Observe a Figura 15.

Figura 15 - Matriz de classificação



Fonte: Próprio autor.

Utilizando a classe *Recommender,* também presente no pacote *recommenderlab*, desenvolve-se um modelo de recomendação com base na matriz

de classificação gerada anteriormente. Esse modelo servirá para nortear as recomendações, juntamente com a matriz de classificação.

Por fim, utilizando a classe *predict*, às recomendações são geradas para cada usuário. Como citado anteriormente, a classe utiliza o modelo de recomendação e a matriz de classificação na tentativa de recomendar artistas para usuários. Na Tabela 5 é possível observar as recomendações geradas pelo sistema para o usuário *7687*.

Tabela 5 - Recomendação musical gerada

Usuário	Recomendações
7687	Pink Floyd
	Black Sabbath
	Judas Priest
	Led Zeppelin
	Xandria

Fonte: Próprio autor.

4.5. Geração do grafo

A linguagem escolhida para se desenvolver o grafo foi Java. Já o ambiente de desenvolvimento utilizado foi o NetBeans. As bibliotecas utilizadas foram *java.io.BufferedReader, java.io.File, java.io.FileReader, java.io.IOException, java.nio.file.Files* e *java.util.ArrayList.*

A maioria das bibliotecas utilizadas o foram pela necessidade de carregar arquivos na memória, tal como a lista com as recomendações para cada usuário. Entretanto, surgiu a necessidade da criação de diversos arquivos, tendo em vista que as recomendações foram dadas por meio de códigos. Por exemplo, caso o sistema desenvolvido em R recomendar a banda de rock *AC/DC*, a recomendação não viria com o nome do grupo propriamente dito, mas com o código *279.*

Inicialmente o grafo, representado por uma matriz de adjacência, foi criado apenas relacionando o gênero musical com seu respectivo subgênero. Essa informação foi retirada da base de dados fornecida pela DBpedia. Durante a

montagem da matriz de adjacência e posteriormente, a geração da justificativa baseada nas recomendações e na matriz, a relação entre gênero e subgênero se mostrou insuficiente, dado que utilizando apenas o subgênero como ponte de ligação, o grafo ficou esparso e, consequentemente, não foi possível aumentar a relação entre gêneros.

Tendo em vista esse problema, houve a necessidade de localizar mais informações que conseguissem relacionar o gênero musical. Com isso, na base de dados que traz informações a respeito dos gêneros, foram escolhidos outros dois atributos que relacionassem o gênero de alguma forma: origem estilística e instrumentos utilizados pelos gêneros. Além disso, foi decidido acrescentar o gênero como atributo relevante para a recomendação. A ordem de relevância foi definida da seguinte forma: o gênero prevalece sobre o subgênero, origem estilística e instrumentos, subgênero prevalece sobre origem estilística e instrumentos e origem estilística prevalece sobre instrumentos. Observe a Tabela 6.

Tabela 6 - Atributos das bases de dados extraídas da DBpedia

Base de dados	Atributos
Gêneros Musicais	Gênero, subgênero, instrumentos e origem estilística
Cantores	Nome e gênero musical
Bandas	Nome e gênero musical

Fonte: próprio autor.

No preenchimento da matriz de adjacência que iria representar o grafo, o algoritmo realiza uma análise para conseguir localizar gêneros e seus respectivos subgêneros. Caso o subgênero fosse localizado, a matriz era preenchida com o valor 1, indicando que o gênero da coluna era subgênero da linha.

Aplicando previamente a ordem de relevância descrita acima, ao se localizar um gênero que utilizasse algum instrumento que outro gênero musical também utilizasse, a matriz de adjacência era preenchida com o valor 2. Além disso, o algoritmo verificava se o campo que seria preenchido já não estaria com o valor 1.

Caso o campo esteja preenchido, o algoritmo continuava a vasculhar com o objetivo de encontrar a relação entre os gêneros com base nos instrumentos.

O mesmo ocorre quando o algoritmo busca gêneros com origens estilísticas em comum. Ao localizar, o campo era preenchido com o valor 3, a menos que já estivesse preenchido com 2 ou 1.

4.6. Esquema de dados

Como citado anteriormente, foi necessário a geração de diversos arquivos. Os arquivos criados o foram, tendo em vista a conexão de informações dispersas entre as bases de dados. Os arquivos foram distribuídos da seguinte forma: o arquivo *recomendacoes* são as recomendações geradas pelo sistema desenvolvido em R, separadas por usuário. Cada usuário recebeu 5 recomendações. Esse arquivo possui relação direta com o arquivo *codigoArtistas* que, por sua vez, possui as informações de código de identificação e nome do artista. A relação entre esses dois arquivos se dá no intuito de localizar o nome propriamente dito do artista.

O arquivo *dadosArtistasBandasGeneros* traz o nome do artista ou banda e respectivamente, seu gênero musical. Com o nome do artista coletado anteriormente no arquivo *codigoArtistas*, é possível identificar o gênero de todos os artistas ouvidos e recomendados, para se montar o grafo.

Para se coletar informações sobre os gêneros, foi necessário criar os arquvios *generosFinal* e *dadosSobreGeneros.* Esses arquivos possuem informações como o subgênero do gênero do artista, bem como sua origem estilística e instrumentos utilizados pelo gênero. Esses dados foram fundamentais para a criação do grafo e geração das justificativas.

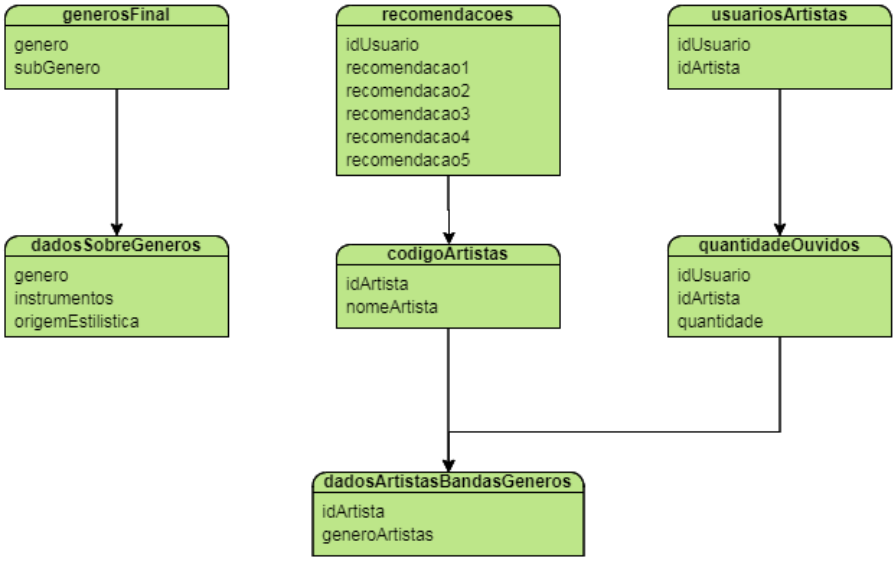
Já o arquivo *usuariosArtistas* possui o código de identificação do usuário e do artista. Nesse arquivo, as informações se repetem diversas vezes. Cada vez que a informação se repete (por exemplo, diversas linhas com o mesmo código de usuário e artista), indica que o usuário ouviu o mesmo artista mais de uma vez.

Por sua vez, o arquivo *quantidadeOuvidos* tem por informação o código do usuário, do artista e a quantidade de vezes que cada usuário ouviu determinado artista. Essa base possui relação também com o *codigoArtistas* com o intuito de identificar o nome do artista e posteriormente, seu gênero. Ela também tem relação

com a base *usuariosArtistas*. Por meio dela foi possível determinar a quantidade de vezes que cada usuário ouviu cada artista.

Considerando que nem todo arquivo gerado teve relação direta com todos os arquivos, a Figura 16 a seguir auxilia no entendimento da relação entre os arquivos de dados.

Figura 16 - Esquema de dados



Fonte: próprio autor.

Para se construir a justificativa da recomendação, foi necessário carregar no Java cada arquivo citado acima. Pensou-se em gerar um único arquivo com todas as informações ou, ao menos, tentar unificar algumas informações com o objetivo de reduzir a quantidade de arquivos. Porém, tendo em vista a repetição de informações e a dificuldade em localizá-las posteriormente, optou-se pela divisão dos dados.

4.6.1. Geração das justificativas

Com a posse do grafo, das recomendações e com o esquema de dados, era preciso então gerar as justificativas diante das recomendações dadas. É preciso levar em consideração que a identificação do usuário é por meio de código, bem como a do artista que o usuário ouviu.

Inicialmente, armazena-se a identificação do usuário, coletando a informação da matriz de recomendações. A matriz de recomendações possui o código do usuário e o código de cinco artistas que foram recomendados para aquele usuário.

Em um vetor, também foram armazenadas as recomendações para o usuário em questão.

O próximo passo é localizar o nome do artista de cada recomendação, dado que as recomendações são dadas por meio de código. Essa informação é guardada no vetor criado anteriormente para se armazenar o código de cada artista recomendado.

Na sequência, com o identificador do usuário armazenado em uma variável anteriormente, o algoritmo procura por artistas que o usuário ouviu e os armazena em um array. Os artistas são armazenados em ordem do mais ouvido para o menos ouvido pelo usuário.

A etapa a seguir consiste na localização do gênero do artista recomendado, bem como o gênero do artista ouvido. Com a posse dessa informação, inicialmente o algoritmo verifica se os gêneros musicais dos artistas (tanto o recomendado quanto o ouvido) são idênticos. Caso não sejam, o algoritmo vasculha a matriz em busca de relações entre o gênero musical do artista recomendado e o gênero musical do artista ouvido.

As informações são armazenadas em um Array de Strings, que deverá ser composta pelo tipo de justificativa (0 para gêneros idênticos, 1 para relação de gênero e subgênero, 2 para relação de instrumentos semelhantes entre gêneros e 3 para origens estilísticas idênticas), o nome do artista recomendado, o nome do artista ouvido, o rank do artista ouvido entre as preferências do usuário e o código identificador do usuário.

O sistema repete esse procedimento para cada recomendação que foi gerada. Tendo em vista que a quantidade de usuários é de 997, o sistema realiza essa iteração 4985 vezes.

Visto os meios utilizados para se alcançar o objetivo geral desta pesquisa, o capítulo que se segue irá expor os resultados obtidos com o algoritmo desenvolvido, bem como exemplos do que é apresentado pelo algoritmo.

5. RESULTADOS

Durante a análise dos resultados do algoritmo, notou-se que o código não foi capaz de gerar uma justificativa para todas as recomendações. Ao analisar os arquivos coletados do site DBpedia, observou-se que nem todo artista que estava presente na base coletada do trabalho de Schedl estava presente na base coletada da DBpedia. Tendo em vista esse problema, em alguns casos, o algoritmo não conseguiu relacionar o gênero do artista recomendado com o gênero do artista ouvido.

Apesar disso, o algoritmo conseguiu justificar 4722 recomendações de um total de 4985, uma taxa de aproximadamente 94,72% de justificativas para a amostragem de 997 usuários.

Observe a Tabela 7 a seguir, que descreve as variações no tocante as justificativas.

Tabela 7 - Tipos de justificativas

Código	Descrição
0	Indica que o gênero do artista ouvido é o mesmo que o do artista recomendado
1	Indica que o gênero do artista recomendado é subgênero do gênero do artista ouvido
2	Indica que os instrumentos utilizados no gênero do artista ouvido são semelhantes aos utilizados no gênero do artista recomendado
3	Indica que a origem estilística do gênero do artista ouvido é a mesma que a do gênero do artista recomendado

Fonte: próprio autor.

Para um melhor entendimento da forma como as justificativas são apresentadas, observe a Tabela 8 a seguir.

Tabela 8 - Estrutura das justificativas

Tipo de informação	Descrição
Tipo de Recomendação	Varia entre 0 e 3 e indica qual a relação que o gênero do artista ouvido possui com o gênero do artista recomendado
Nome Artista Recomendado	Informa o nome do artista que foi recomendado ao usuário em questão
Nome Artista Ouvido	Informa o nome do artista ouvido pelo usuário em questão, no qual o algoritmo conseguiu encontrar relação com o artista recomendado
Rank Artista Ouvido	Informa a classificação do artista ouvido diante da quantidade de vezes que o usuário em questão o ouviu, comparado aos demais artistas ouvidos
Código Usuário	Informa o código do usuário no qual a justificativa está sendo estruturada

Fonte: próprio autor.

Note que a justificativa é montada inicialmente com o tipo de recomendação, que varia entre 0 e 3, seguido do nome do artista recomendado, nome do artista ouvido, rank do artista ouvido e o código do usuário. Essas informações são suficientes para se gerar uma justificativa ao usuário, dado que com isso é possível identificar a origem da recomendação e relacionar o gênero do artista ouvido com o gênero do artista recomendado, além de classificar o artista recomendado, o que dá ainda mais credibilidade a justificativa.

Por meio do grafo, o software consegue identificar diversas relações entre os gêneros. Porém, a justificativa é armazenada com base na classificação do artista mais ouvido: ao se deparar com várias justificativas para a mesma recomendação, a justificativa que será apresentada ao usuário será aquela que contém o artista mais ouvido pelo usuário. Para o usuário 7687, quase todas as justificativas geradas foram com base nos instrumentos semelhantes, pois o sistema classificou as justificativas de acordo com o artista que o usuário ouviu com maior frequência. Observe a Figura 17 e o APÊNDICE A.

Figura 17 - Justificativas geradas para o usuário 7687

Tipo de Recomendação	Nome Artista Recomendado	Nome Artista Ouvido	Rank Artista Ouvido	Código Usuário
2	Pink Floyd	Foo Fighters	1	7687
2	Black Sabbath	Foo Fighters	1	7687
2	Judas Priest	Foo Fighters	1	7687
2	Led Zeppelin	Foo Fighters	1	7687
3	Xandria	Foo Fighters	1	7687

Fonte: próprio autor.

Porém, em alguns casos, como mencionado anteriormente, o sistema não foi capaz de relacionar o gênero do artista recomendado com o gênero do artista ouvido.

No exemplo em questão, o artista recomendado foi a banda *Skybreed.* O algoritmo não foi capaz de relacionar os gêneros dos artistas ouvidos com o gênero do artista recomendado, pois as bases da DBpedia não possuíam informações sobre o artista em questão. Observe a Figura 18 seguir, com a justificativa das recomendações para o usuário 2465033. Observe que a terceira justificativa está nula.

Figura 18 - Justificativas geradas para o usuário 2465033

Tipo de Recomendação	Nome Artista Recomendado	Nome Artista Ouvido	Rank Artista Ouvido	Código Usuário
2	Pink Floyd	Jens Lekman	1	2465033
2	In Flames	The Olivia Tremor Control	7	2465033
NULL				
2	Oomph!	Jens Lekman	1	2465033
2	Lacuna Coil	Explosions in the Sky	4	2465033

Fonte: próprio autor.

Em alguns casos, o algoritmo não conseguiu relacionar os gêneros dos artistas, o que também gerou justificativas sem informações. No caso da Figura 19, que apresenta a justificativa para as recomendações do usuário 2949370, para o artista *Andrea Berg,* o software não conseguiu relacionar os gêneros.

Figura 19 - Justificativas geradas para o usuário 2949370

Tipo de Recomendação	Nome Artista Recomendado	Nome Artista Ouvido	Rank Artista Ouvido	Código Usuário
Null				
2	Star Guard Muffin	U2	7	2949370
2	Jay Sean	Red Hot Chili Peppers	2	2949370
2	Oomph!	Red Hot Chili Peppers	2	2949370
2	Lacuna Coil	Red Hot Chili Peppers	2	2949370

Fonte: próprio autor.

Exposto os resultados obtidos com o algoritmo, o capítulo de conclusão abordará algumas considerações sobre a pesquisa, bem como suas contribuições e sugestões para projetos futuros.

6. CONCLUSÃO

Este capítulo apresenta as considerações finais sobre o trabalho desenvolvido, bem como suas contribuições, possíveis melhorias e pesquisas futuras.

6.1. Considerações finais

O trabalho desenvolvido teve como objetivo auxiliar no aprimoramento da interpretabilidade de sistemas de recomendação. A interpretabilidade mostra grande importância diante do cenário atual, levando em consideração que os sistemas de recomendação estão cada vez mais presentes no cotidiano da sociedade.

O processo para se chegar a esta conclusão passou por diversas etapas. Foram apresentados os principais tipos de sistemas de recomendação, além de como se iniciou o projeto LOD, sua estrutura, variações, a forma como os dados devem ser publicados e suas principais contribuições.

Além disso, foi mostrado também a importância da interpretabilidade em sistemas de recomendação na seção Interpretabilidade sobre decisões. Diante da presença cada vez maior desses sistemas em áreas importantes, a utilização da interpretabilidade é inquestionável.

Diante dessas questões, o algoritmo teve um rendimento de 94,72% de justificativas geradas e mostrou-se satisfatório para o objetivo da pesquisa.

6.2. Contribuições

Por meio da pesquisa e desenvolvimento do sistema de recomendação, foi possível contribuir com um novo método para gerar justificativas em sistemas de recomendação. A pesquisa focou em sistemas de recomendação musical, porém os conhecimentos presentes neste trabalho são transferíveis para outras áreas que utilizam sistemas de recomendação.

Foi exposto também quão grande é o leque de possibilidades em se utilizar a DBpedia. A grande quantidade de dados disponibilizada pelo projeto pode auxiliar em diversas pesquisas.

Além disso, foi exposta a importância da utilização da interpretabilidade em sistema de recomendação. Sua utilização é de grande relevância, especialmente diante de situações que afetam diretamente a vida de usuários ou de terceiros.

6.3. Sugestões para trabalhos futuros

Tendo em vista o quão abrangente é a área de sistemas de recomendação e afins, alguns trabalhos podem ser desenvolvidos para melhoria do projeto atual, bem como no desenvolvimento de outros projetos relacionados.

Um dos principais aprimoramentos a serem feitos no software envolve as bases de dados, tendo em vista que devido a ausência de algumas informações nas bases extraídas da DBpedia, o software não conseguiu gerar justificativas para todas as recomendações. Um provável trabalho a ser realizado é complementar a base de dados atual, utilizando outras bases de dados que forneçam as informações que faltam na base fornecida pela DBpedia

Outra melhoria seria a otimização no processo de geração das justificativas, tendo em vista que, para a amostragem atual, o software demorou em média uma mora para se gerar todas as justificativas. Possivelmente, a quantidade de arquivos gerados para se montar a justificativa seria reduzido, com o objetivo de otimizar o processo. Isso diminuiria a quantidade de laços de repetição, acarretando em uma diminuição de uso de memória e redução no processamento dos dados.

Ainda relacionado ao projeto atual, é importante realizar uma validação das recomendações geradas, o que avaliaria sua qualidade. Esse processo seria realizado com um grupo de usuários reais, com o objetivo de avaliar se as recomendações, bem como as justificativas são plausíveis. A informação seria apresentada aos usuários, que iriam relatar se aquela recomendação, aliada a justificativa teria, de fato, sentido para suas preferências musicais.

No que envolve outros projetos, utilizar os conceitos expostos neste trabalho sobre interpretabilidade em sistemas de recomendação se mostra crucial para a criação e aperfeiçoamento de sistemas de recomendação, além de aumentar a confiabilidade em recomendações geradas. Um exemplo inquestionável da aplicação da interpretabilidade, como já mencionado anteriormente, é o algoritmo COMPAS, levando em consideração que o algoritmo está atualmente afetando a vida de usuários e terceiros de forma substancial.

REFERÊNCIAS

ADADI, Amina; BERRADA, Mohammed. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). **IEEE Access,** v. 6, p. 52138-52160, 2018.

ADOMAVICIUS, Gediminas; TUZHILIN, Alexander. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. **IEEE transactions on knowledge and data engineering**, v. 17, n. 6, p. 734-749, 2005.

AMARAL, Adriana; AQUINO, Maria Clara. **Práticas de folksonomia e social tagging no Last. fm.** In: Proceedings of Conference on Human Factors in Computing Systems. 2008.

AUER, Sören; BIZER, Christian; KOBILAROV, Georgi; LEHMANN, Jens; CYGANIAK, Richard; IVES, Zachary. Dbpedia: A nucleus for a web of open data. In: **The semantic web.** Springer, Berlin, Heidelberg, 2007. p. 722-735.

BERNERS-LEE, Tim; CHEN, Yushin; CHILTON, Lydia; CONNOLLY, Dan; DHANARAJ, Ruth; HOLLENBACH, James; LERER, Adam; SHEETS, David. Tabulator: Exploring and analyzing linked data on the semantic web. In: **Proceedings of the 3rd international semantic web user interaction workshop.** 2006. p. 159.

BIZER, Chris; CYGANIAK, Richard; HEATH, Tom. **How to publish linked data on the web**. 2007.

BIZER, Christian; HEATH, Tom; IDEHEN, Kingsley; BERNERS-LEE, Tim et. al. Linked data on the web (LDOW2008). In: **Proceedings of the 17th international conference on World Wide Web.** ACM, 2008. p. 1265-1266.

BIZER, Christian; HEATH, Tom; BERNERS-LEE, Tim. Linked data: The story so far. In: **Semantic services, interoperability and web applications: emerging concepts**. IGI Global, 2011. p. 205-227.

BIZER, Christian; LEHMANN, Jens; KOBILAROV, Georgi; AUER, Sören; BECKER, Christian; CYGANIAK, Richardl. DBpedia-A crystallization point for the Web of Data. **Web Semantics: science, services and agents on the world wide web**, v. 7, n. 3, p. 154-165, 2009.

CANALTECH. **Como o Facebook sugere amigos que você conhece offline?** Disponível em: encurtador.com.br/dfLNT. Desde 08 de novembro de 2017. Acesso em 30 de novembro de 2020.

CARVALHO, Diogo V.; PEREIRA, Eduardo M.; CARDOSO, Jaime S. Machine learning interpretability: A survey on methods and metrics. **Electronics**, v. 8, n. 8, p. 832, 2019.

CAZELLA, Sílvio César; NUNES, M. A. S. N.; REATEGUI, Eliseo. A Ciência da Opinião: Estado da arte em Sistemas de Recomendação. **André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski..(Org.). Jornada de Atualização de Informática-JAI,** p. 161-216, 2010.

CAZELLA, Sílvio César; REATEGUI, Eliseo Berni; MACHADO, Munique; BARBOSA, Jorge Luis V. Recomendação de objetos de aprendizagem empregando filtragem colaborativa e competências. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE).** 2009.

CAZELLA, Sílvio César; REATEGUI, Eliseo Berni. Sistemas de recomendação. In: **São Leopoldo: XXV Congresso da Sociedade Brasileira de Computação.** 2005.

CORBY, Olivier; DIENG, Rose; HÉBERT, Cédric. A conceptual graph model for w3c resource description framework. In: **International conference on conceptual structures.** Springer, Berlin, Heidelberg, 2000. p. 468-482.

COSTA, Antonio Alexandre Moura. **Uma abordagem centrada na filtragem colaborativa para redução do custo computacional do método k- Nearest Neighbors.** 2014. 91f. (Dissertação de Mestrado em Ciência da Computação) Programa de Pós-graduação em Ciência da Computação, Centro de Engenharia

Elétrica e Informática, Universidade Federal de Campina Grande - Paraíba - Brasil, 2014.

DBPEDIA. **ABOUT.** Disponível em: https://wiki.dbpedia.org/about. Acesso em 28 de junho de 2020.

DBPEDIA. **DBPEDIA.** 2019. Disponível em: https://wiki.dbpedia.org/. Acesso em 22 maio 2019.

DOSHI-VELEZ, Finale; KIM, Been. Towards a rigorous science of interpretable machine learning. **arXiv preprint arXiv:1702.08608,** 2017.

DZIEKANIAK, Gisele Vasconcelos; KIRINUS, Josiane Boeira. Web semântica 10.5007/1518-2924.2004 v9n18p20. **Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação**, v. 9, n. 18, p. 20-39, 2004.

GOLDBERG, David; NICHOLS, David; OKI, Brian M.; TERRY, Douglas. Using collaborative filtering to weave an information tapestry. **Communications of the ACM**, v. 35, n. 12, p. 61-71, 1992.

HAHSLER, Michael. **recommenderlab: A framework for developing and testing recommendation algorithms.** 2015.

HAHSLER, Michael. **recommenderlab: Lab for Developing and Testing Recommender Algorithms.** 2017. R package version 0.2-2. Disponível em: http://lyle.smu.edu/IDA/recommenderlab/.

HAHSLER, Michael; VEREET, Bregt; HAHSLER, Maintainer Michael. Package 'recommenderlab'. 2020.

HEATH, Tom; BIZER, Christian. Linked data: Evolving the web into a global data space. **Synthesis lectures on the semantic web: theory and technology**, v. 1, n. 1, p. 1-136, 2011.

IBM. **Sistemas de Recomendação.** 2014. Disponível em: encurtador.com.br/exL09. Acesso em 24 maio 2019.

JAIN, Prateek; HITZLER, Pascal; SHETH, Amit P.; VERMA, Kunal; YEH, Peter Z. et. al. Ontology alignment for linked open data. In: **International semantic web conference.** Springer, Berlin, Heidelberg, 2010. p. 402-417.

LARSON, Jeff; MATTU, Surya; KIRCHNER, Lauren. ANGWIN, Julia. **How We Analyzed the COMPAS Recidivism Algorithm.** ProPublica. 23 de maio de 2016. Disponível em: encurtador.com.br/sHTY0. Acesso em 18 de setembro de 2020.

LAST.FM. **SOBRE A LAST.FM.** Disponível em: https://www.last.fm/pt/about. Acesso em 01 de maio de 2020.

LEHMANN, Jens; ISELE, Robert; JAKOB, Max; JENTZSCH, Anja; KONTOKOSTAS, Dimitris; MENDES, Pablo N.; HELLMANN, Sebastian; MORSEY, Mohamed; KLEEF, Patrick van; AUER, Sören; BIZER, Christian. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. **Semantic Web**, v. 6, n. 2, p. 167-195, 2015.

LIAO, Chih-Lun; LEE, Shie-Jue. A clustering based approach to improving the efficiency of collaborative filtering recommendation. **Electronic Commerce Research and Applications**, v. 18, p. 1-9, 2016.

LOD-CLOUD. **The Linked Open Data Cloud.** 2019. Disponível em: https://lod-cloud.net/. Acesso em 23 maio 2019.

MAYBIN, Simon. **Sistema de algoritmo que determina pena de condenados cria polêmica nos EUA.** BBC News | Brasil. 31 de outubro de 2016. Disponível em: https://www.bbc.com/portuguese/brasil-37677421. Acesso em 18 de setembro de 2020.

MCBRIDE, Brian. The resource description framework (RDF) and its vocabulary description language RDFS. In: **Handbook on ontologies.** Springer, Berlin,

Heidelberg, 2004. p. 51-65.

MICHAEL HAHSLER (2020). **recommenderlab:** Lab for Developing and Testing Recommender Algorithms. R package version 0.2-6. https://github.com/mhahsler/recommenderlab.

MILLER, Tim. Explanation in artificial intelligence: Insights from the social sciences. **Artificial Intelligence**, 2018.

MOLNAR, Christoph et. al. Interpretable machine learning: A guide for making black box models explainable. **Christoph Molnar, Leanpub,** 2018.

NETFLIX. **Como funciona o sistema de recomendações da Netflix.** 2019. Disponível em: https://help.netflix.com/pt/node/100639. Acesso em 2 junho 2019.

OLIVEIRA, Ricardo Santos de. **Diversificação em Sistemas de Recomendação Utilizando uma Abordagem Baseada em Aspectos.** Tese de Doutorado (Tese de Doutorado em Engenharia Elétrica e Informática) - UFCG. Campina Grande - PB. 2020.

PARK, Youngki; PARK, Sungchan; JUNG, Woosung; LEE, Sang-goo. Reversed CF: A fast collaborative filtering algorithm using a k-nearest neighbor graph. **Expert Systems with Applications**, v. 42, n. 8, p. 4022-4028, 2015.

REZENDE, Flávia; BARROS, Suzana de Souza. Discussão e reestruturação conceitual através da interação de estudantes com as visitas guiadas do sistema F&M. **Revista Brasileira de Pesquisa em Ensino de Ciências.** ABRAPEC. v.1, n.2, Maio/Ago. 2001. Disponível em: http://revistas.if.usp.br/rbpec/article/view/203/187. Acesso em 31 setembro 2019.

RIBEIRO, Nuno Magalhães; GOUVEIA, Luis Borges. Proposta de um modelo de referência para as tecnologias multimédia. Portugal: **Revista da Faculdade de Ciência e Tecnologia,** v. 1, p. 109-115, 2004. Disponível em: http://cerem.ufp.pt/~nribeiro/publicacoes/nribeiro_lmbg_tecmm.pdf. Acesso em: 31

setembro 2019.

RICCI, Francesco; ROKACH, Lior; SHAPIRA, Bracha. Introduction to recommender systems handbook. In: **Recommender systems handbook**. Springer, Boston, MA, 2011. p. 1-35.

R PROJECT. **What is R?** Disponível em: https://www.r-project.org/about.html. Acesso em 01 de maio de 2020.

RSTUDIO. **About RStudio.** Disponível em: https://rstudio.com/about/. Acesso em 01 de maio de 2020.

R-BLOGGERS. **Recommender Systems 101 – a step by step practical example in R.** Disponível em: encurtador.com.br/lqAFL. Acesso em 23 de junho de 2020.

SANTOS, Anderson Pimentel dos. **Sistema de recomendação baseado em agrupamento usando Propagação de Afinidades**. 2017. 44 f. Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas, Manaus, 2017.

SEGUNDO, José Eduardo Santarém. **Web semântica:** introdução a recuperação de dados usando SPARQL. 2017.

SCHAFER, J. Ben; KONSTAN, Joseph A.; RIEDL, John. E-commerce recommendation applications. **Data mining and knowledge discovery**, v. 5, n. 1-2, p. 115-153, 2001.

SCHEDL, Markus. **The lfm-1b dataset for music retrieval and recommendation.** In: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval. 2016. p. 103-110.

SHANI, Guy; GUNAWARDANA, Asela. Evaluating recommendation systems. In: **Recommender systems handbook.** Springer, Boston, MA, 2011. p. 257-297.

VIEIRA, Felipe José Rocha; NUNES, Maria Augusta Silveira Netto. Dica: Sistema

de recomendação de objetos de aprendizagem baseado em conteúdo. **Scientia Plena,** v. 8, n. 5, 2012.

VIEIRA, Ronaldo da Mota. *World Wide Web:* terra encantada onde tudo se encontra?. **Educação, Gestão e Sociedade: revista da Faculdade Eça de Queirós,** v. 13, n. 16, 2014.

WEXLER, Rebeca. **When a Computer Program Keeps You in Jail.** The New York Times. 13 de junho de 2017. Disponível em: encurtador.com.br/ftFNS. Acesso em 22 de setembro de 2020.

WHITEHEAD, Jim. **Oralidade e hipertexto:** uma entrevista com Ted Nelson. In. O hipertexto em tradução. RIBEIRO, Ana Elisa; COSCARELLI, Carla Viana (Org.). Belo Horizonte: FALE/UFMG, 2007.

WICKHAM, Hadley; AVERICK, Mara; BRYAN, Jennifer; CHANG, Winston; MCGOWAN, Lucy D'Agostino; FRANÇOIS, Romain; GROLEMUND, Garrett; HAYES, Alex; HENRY, Lionel; HESTER, Jim; KUHN, Max; PEDERSEN, Thomas Lin; MILLER, Evan; BACHE, Stephan Milton; MÜLLER, Kirill; OOMS, Jeroen; ROBINSON, David; SEIDEL, Dana Paige; SPINU, Vitalie; TAKAHASHI, Kohske; VAUGHAN, Davis; WILKE, Claus; WOO, Kara; YUTANI, Hiroaki. **Welcome to the Tidyverse. Journal of Open Source Software**, v. 4, n. 43, p. 1686, 2019.

WIKIPÉDIA. **WIKIPÉDIA.** 2019. Disponível em: https://www.wikipedia.org/. Acesso em 22 maio 2019.

W3. **Linked Data.** 2006. Disponível em: https://www.w3.org/DesignIssues/LinkedData.html. Acesso em 19 maio 2019.

W3C. **Resource Description Framework: Concepts and Abstract Syntax.** 2004. Disponível em: https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/. Acesso em 26 maio 2019.

APÊNDICE A – Arquivo de justificativas do algoritmo

```
2;Pink Floyd;Foo Fighters;1;7687
2;Black Sabbath;Foo Fighters;1;7687
2;Judas Priest;Foo Fighters;1;7687
2;Led Zeppelin;Foo Fighters;1;7687
3;Xandria;Foo Fighters;1;7687

2;Be'lakor;J Mascis;74;1029562
2;Children of Bodom;The Beatles;1;1029562
2;McFly;The Beatles;1;1029562
1;Jonas Brothers;The Beatles;1;1029562
0;One Direction;The Beatles;1;1029562

2;McFly;The Doors;2;2168899
2;Jonas Brothers;The Doors;2;2168899
2;Sarah Brightman;Opeth;1;2168899
2;One Direction;Opeth;1;2168899
2;Madball;Porcupine Tree;3;2168899
```

Fonte: próprio autor.